ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Μηχανική μάθηση για γεωφυσικές εφαρμογές

Φοιτήτρια: ΜΑΡΙΑΝΘΗ ΑΔΑΜΟΠΟΥΛΟΥ-ΣΟΥΛΑΝΤΙΚΑ
ΑΜ: 50106941

Επιβλέπων Καθηγητής

ΓΕΩΡΓΙΟΣ ΧΛΟΥΠΗΣ
Αναπληρωτής Καθηγητής

Συνεπιβλέπων Καθηγητής

ΠΑΝΑΓΙΩΤΗΣ ΦΩΤΟΠΟΥΛΟΣ
Αναπληρωτής Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΙΟΥΛΙΟΣ 2023

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

Machine Learning for geophysical applications

Student: MARIANTHI ADAMOPOULOU-SOULANTIKA
Registration Number: 50106941

Supervisor

GEORGE HLOUPIS
Associate Professor

Co-supervisor

PANAGIOTIS PHOTOPOULOS
Associate Professor

ATHENS-EGALEO, JULY 2023

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

(Ονοματεπώνυμο), (βαθμίδα)	(Ονοματεπώνυμο), (βαθμίδα)	(Ονοματεπώνυμο), (βαθμίδα)
Panagiotis Fotopoulos Digitally signed by Panagiotis Fotopoulos Date: 2023.07.19 18:12:11 +03'00' (Υπογραφή)	Digitally signed by Konstantinos Moutzouris Date: 2023.07.19 09:54:06 +03'00' (Υπογραφή)	Georgios Chloupis Digitally signed by Georgios Chloupis Date: 2023.07.25 21:59:43 +03'00' (Υπογραφή)

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η Αδαμοπούλου-Σουλαντίκα Μαριάνθη του Χρήστου, με αριθμό μητρώου 50106941 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι 14/07/2023 και έπειτα από αίτησή μου στη Βιβλιοθήκη και έγκριση του επιβλέποντος/ουσας καθηγητή/ήτριας.»

Ο/Η Δηλών/ούσα
(Ονοματεπώνυμο φοιτητή/ήτριας)
Αδαμοπούλου-Σουλαντίκα Μαριάνθη

(Υπογραφή φοιτητή/ήτριας)

Περίληψη

Τα τελευταία χρόνια, η Βαθιά Μάθηση έχει συναντήσει μεγάλο ενδιαφέρον στην γεωφυσική κοινότητα. Εφαρμόζεται σε πολλές γεωφυσικές εφαρμογές, καθιστώντας ευκολότερη την επίλυση προβλημάτων. Αντιμετωπίζει προβλήματα μεγάλης κατανάλωσης υπολογιστικών πόρων και χρόνου, για μεγάλο όγκο δεδομένων καθώς και την κατάρα της διαστατικότητας. Με τα χρόνια, έχει υπάρξει πρόοδος στις μεθόδους της. Ένα πλεονέκτημα που παρέχει η χρήση της, είναι ότι επιτρέπει στους χρήστες της την χρήση τηλεσκόπισης, για την απόκτηση και ανάλυση δεδομένων ακόμα και από δύσβατες για τον άνθρωπο περιοχές. Η Βαθιά Μάθηση με την βοήθεια των βιβλιοθηκών TensorFlow και Keras, έχει λάβει μέρος στην βελτίωση της ανακάλυψης του περιβάλλοντα χώρου.

Λέξεις – κλειδιά

Μηχανική Μάθηση, Βαθιά Μάθηση, Σεισμοί, TensorFlow και Keras βιβλιοθήκες, γεωφυσική, μοντέλα βαθιάς μάθησης.

Abstract

In recent years, deep learning has drawn a lot of interest from the geophysical community. It can be used for a variety of geophysical employments, by providing simpler and easier solutions to problems. It addresses problems to do with excessive need of computational resources and time consumption from large amounts of data, as well as the curse of dimensionality. Over the years there has been great progress with Deep Learning methods. An advantage of using Deep Learning in certain applications is that it allows its users to use remote sensing, so they can collect and analyze data even from places that are difficult to be approached by humans. With the help of TensorFlow and Keras libraries, Deep Learning plays a part in the improvement of exploring surrounding environments.

Keywords

Machine Learning, Deep Learning, Earthquakes, libraries TensorFlow and Keras, geophysics, DL models.

Contents

List of Table

List of Images

Alphabetical Index

ML: Machine Learning

DL: Deep Learning

DNN: Deep Neural Network
RNN: Recurrent Neural Network

CNN: Convolutional Neural Network

ANN: Artificial Neural Network

NNA: Neural Network Architecture

EEW: Earthquake Early Warning
STA: Short-Term Average
LTA: Long-Term Average
SNR: Signal-to-Noise Ration
SVM: Support Vector Machine
WST: Wavelet Scattering Transform
GNSS: Global Navigation Satellite System
AI: Artificial Inteligence
SNNS: Stuttgart Neural Network Simulator
MILA: Montreal Institute for Learning Algorithms
WEKA: Waikato Environment for Knowledge Analysis
API: Application Programming Interface
MLP: MultiLayer Perceptron
FCNN: Fully Connected Neural Network
SEG: Society of Exploration Geophysics
AGU: American Geophysical Union
STEAD: STanford Earthquake Dataset
ReLU: Rectified Linear Unit
LSTM: Long Short-Term Memory
GRU: Gated Recurrent Unit
FWI: Full Waveform Inversion
CAE: Convolutional AutoEncoder
GAN: Generative Adversarial Networks
MSE: Mean Square Error
CE: Cross-Entropy
SGD: Stochastic Gradient Descent
CRED: CNN-RNN Earthquake Detector
DPP: DeepPhasePick

EQT: EarthQuake Transformer
GPD: Generalized Phase Detection
SCEDC: Southern Californian Earthquake Data Center
STP: Seismic Transfer Program
NEIC: National Earthquake Information Center
NCEDC: Northern Californian Earthquake Data Center
ROC: Receiver's Operating Characteristic
OBC: Ocean Bottom Seismometer

Introduction

This paper discusses the place and the impact of Machine Learning on the field of geophysics, as also the offered facilities that are provided by the use of Deep Learning models. Primarily, the use of algorithm in Python in combination with the Keras and TensorFlow libraries, especially in comparison to traditional methods. In addition, an analysis in the field of seismology is supplied, as well as problems and ways of dealing with them. Also are mentioned cases where Deep Learning manages to give more precise outcomes than those of professionals, but also cases where experts give more accurate outcomes. Then, are presented examples of research on the subject, as well models and datasets.

It is noteworthy to mention that the definition of Machine Learning was first formulated by Samuel[1] , with Mitchell[2] providing a frequently cited definition:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." [2]

Subject of the thesis

The primary focus of this thesis is the deployment of ML in geophysical applications. At the same time it is studied a bit more specifically, the use of python libraries TensorFlow and Keras for the same filed. It's also very important how the use of them has helped and developed today's algorithms of DL, for instance in earthquake detection, even for places that are dangerous and risky for people to visit. It is very interesting what possibilities these technologies give to humans. Since DL has found application in the geophysical domain, has helped humanity to discover more things about its environment. Furthermore, it is very useful on today's problems, as it gives accurate results and in very short time, depending on the amount of the data that have to been processed, but also quicker than a professional. Even though, there are still times that experts give better results than trained models. It is an amazing topic to work on, as it analyzes raw data and gives so many important information to scientist, but also non scientist people. A remarkable outcome of all this process is the possibility to warn early the society about an upcoming earthquake or volcano explosion etc and save people's lives. The application of all this, is crucial for the society.

Purposes and objectives

The ambition of this thesis is to demonstrate ML's role in geoscience community as well its applications and results. In addition, it is to demonstrate the usefulness of DL models, by the use of libraries such as Keras and TensorFlow, for geophysics applications, for instance in seismic data analysis, early warning, or microearthquake detection, etc.

Structure

The first chapter, is a general introduction to the topic, with some information about each field that is discussed in this review. Furthermore, in all chapters, images and tables are included for a better understanding. Chapter 2 talks about ML in geoscience. In addition, Chapter 3 contains more details about DL's implementation in the domain of geoscience, especially earthquakes. Different DL models as well as neural networks are also discussed in this chapter, including some examples for the results to be easier understood. Furthermore, Chapter 4 focuses on earthquakes. It shows the need that exist for the study of this area, as well as includes a brief mention of the role of DL in supporting these studies. Also, it is discussed how these two fields are combined in order to provide improved results. Chapter 5 is where geoscience and IT really meet. In this chapter it is presented how DL is really making a difference in the methods that are followed to process seismic data in comparison to traditional methods. This is something that is somewhat visible in the previous chapters too, especially chapter 4, but here it is strongly presented. Chapter 6 is the summary of this entire review. It is a recapitulation of what has already been discussed in the previous chapters. It consists of the conclusions that all the information lead to. Lastly, chapter 7 is the bibliography of the sources used to provide the necessary information and knowledge to convey the subject of this review to those interested.

1 Geophysics with Machine Learning

Geophysics is the study of Earth's structure and composition from its core to its surface. Nowadays, focus has shifted from just studying Earth to including outer space, as also the upper atmospheres of planets other than Earth and Earth's atmosphere. Common methods used in geophysics are observation, process, modeling and prediction. Observation is the most crucial way that humans can study their surrounding environment and begin forming an understanding of all the yet undiscovered or unexplained mysteries of their planet and solar system. In the core of this science is the application of different mathematical solutions based on the relevant physical laws. One very important parameter is what is called “denoising”, which is a critical step in acquiring significant results after the processing of the collected raw data.[3]

1.1 Earthquake Detection

Investigating seismic wavelengths is one of the most common imaging methods in exploration geophysics, where the Earth's subsurface is depicted and mapped through inverting surface-located physical fields. Reflective seismic waves are employed in seismic exploration to foretell the underlying structures of the planet, after the fundamental procedures for seismic data sampling, processing (such as denoising and interpolation), inversion (such as migration and imaging), and interpretation (such as fault detection and facies categorization). Figure 1 provides a summary of the exploration process, while figure 2 contrasts conventional and DL-based exploration geophysics techniques.[3]

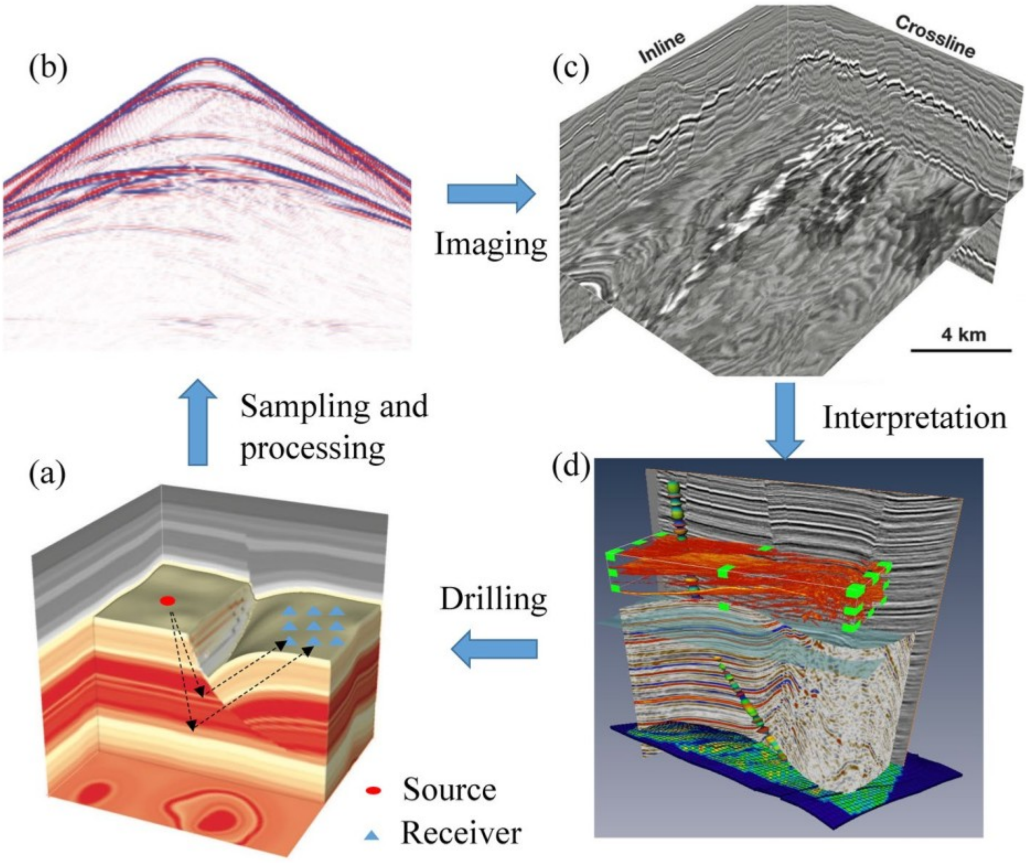


Figure 1 The exploration geophysical procedure. (a) The structures of the subterranean. The seismic wave gets stimulated at sources, as illustrated by red dots, and spreads down way till the reflector before ascending to the receivers, as illustrated by blue dots. (b) The seismic data afterwards been processed. (c) The seismic imaging outcome, with the reflectors been represented by the lines. (d) Underground features are construed to specify the location of the reservoir.[3]

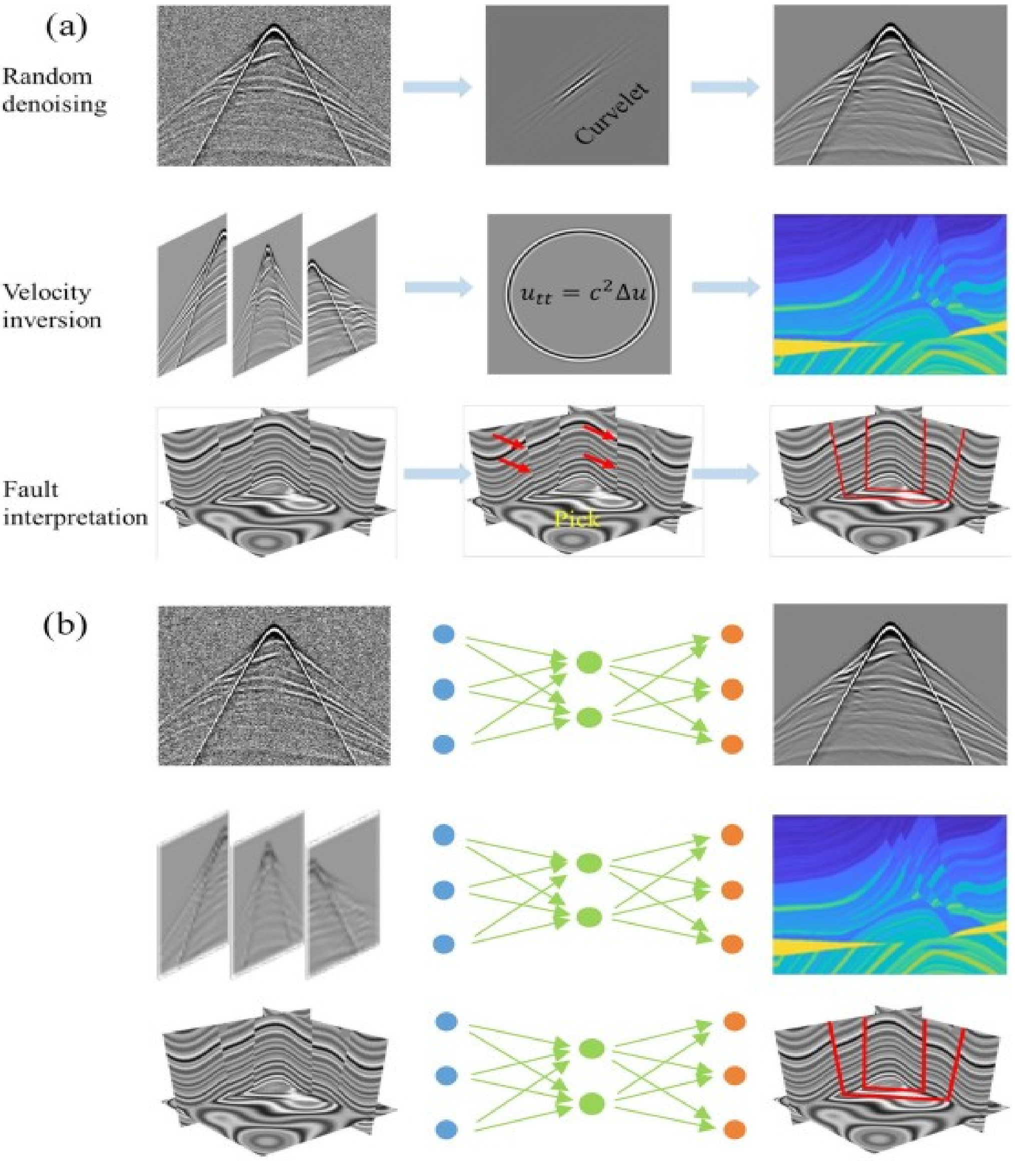


Figure 2 An evaluation of DL-based and conventional geophysics exploration methods. (a)The curvelet denoising method used in random denoising tasks[128] takes into consideration that under a curvelet transform a signal can be sparse, so it is beneficial to use a matching method for denoising. In forward and adjoint modeling,in order to maximize the efficiency of velocity inversion jobs the full waveform is used based on the wave equation. Finally, interpreters choose faults in fault interpretation tasks. (b)Regression issues which are optimized by neural networks, are applicable to previously mentioned tasks. Different neural network topologies could be required for various purposes [3]

The purpose of earthquake data processing differs greatly from the purpose of exploration geophysics, which is why it is relevant to focus upon DL-based seismic signal processing. The classification and arrival picking are two facets of the early processing of seismic signals. Classification separates actual earthquakes from background noise, while arrival picking determines when primary (P) and secondary (S) waves will arrive. Some farther deployments include Earth tomography and the location of earthquakes. DL has displayed encouraging outcomes in various applications.[3][4]

The arrival times of P and S waves are determined by using the aforementioned arrival picking method. Professionally trained humans are capable of providing more accurate results than traditional automated algorithms, as for instance, the automated short-term average (STA) and long-term average (LTA) methods rely on thresholding setting[3]. DL can help with overcoming such weaknesses by using arrival picking to capture a clear image of Earth's structure[5]. One way to achieve significantly more accurate results than offered by the STA/LTA methods, or even human experts, is using large training sets in arrival picking and classification[6][7][8]. In the case of inadequate labels, a labeled data set has the possibility to be artificially expanded with the help of the GAn-based EarthquakeGen model[9]. Performing artificial sampling on the training set can remarkably improve detection accuracy. Meanwhile earthquake detection and phase picking accuracies can be farther progressed by concurenting the both tasks[10].

1.1.1 Neural Networks

The most fundamental but also most difficult tasks, in earthquake early warning, EEW, are earthquake signal and noise classification. False and missed alarms plague traditional EEW systems. Since Deep Neural Network, DNN, is a classification task, it can be directly applied to signal and noise distinguishment. With an adequate training set the precision of DNNs can reach 99.2%[11] and 99.5%[12] for various domains. In order to detect smaller both weaker earthquake signals that are rugged to intense noise and nonseismic signals, a residual network which includes convolutional and recurrent units can be deployed[13]. Another difficult task that Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can be used for is the differentiation of anthropocentric causes, like mining, quarry blasts, and tectonic seismicity[14].. For some particular tasks, such as volcano seismic detection, more categories of signals are necessary[15]. Volcano-seismic monitoring also takes uncertainty into consideration[16].

To give some examples, in a study conducted by Meier et al. (2019) five DNNs were trained to distinguish between seismic signals and noise. The training set consisted of 374,000 earthquake records and 946,000 noise records of three channels[12]. In another study, Li et al. (2018) trained ML algorithms to distinguish between earthquake P waves and local impulsive noise using a large dataset[11]. Furthermore, Linville et al. (2019) employed an RNN and a CNN for the purpose of identifying events as either quarry blasts or earthquakes[14]. Volcanic seismic detection has a similar purpose to EEW, namely to determine the danger levels of an event. In a study by Malfante et al. (2018) a total of 102 characteristics that were derived from the auditory and seismic fields[17], and the classes of long-period occurrences, volcanic tremors, volcano-tectonic occurrences, explosions, hybrid occurrences, and tornadoes were investigated.

A study by Mallat (2012) is one example of classifying earthquakes with restricted training samples, where a support vector machine (SVM) and wavelet scattering transform (WST) were employed[18]. The WST includes consists of a series of wavelet transforms, a module operator, and an averaging operator, which corresponds to the convolutional filters, nonlinear operator, and pooling operator in a CNN. The crucial discrepancy among the WST and a CNN is that in the WST, the filters are preliminary designed using the wavelet transform. In this particular scenario, only 100 records were used for training, while 2000 records were used for testing. Using the WST method, a classification accuracy of 93% was achieved. The architecture of the WST algorithm is presented in figure 3.

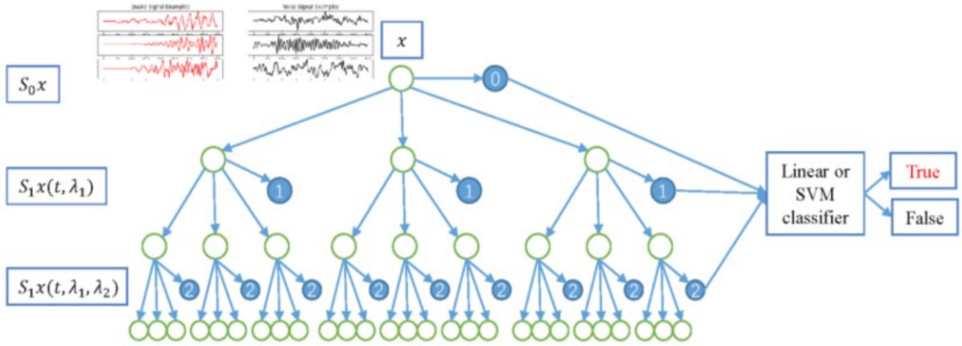


Figure 3 (a) The WST architecture. WST outputs are bundled, as opposed to CNN ones, alongside the outputs of every layer. Subsequently, the WST outputs are used as characteristics of a classifier[4].

1.2 Earthquake Applications

Assessing earthquake location and magnitude is crucial in EEW and subterranean imaging. Contractual earthquake detection is based importantly on a velocity model which suffers from imprecise phase selection.

Zhang et al. (2020) utilised a CNN to locate seismic sources by obtained waveforms at multiple locations[19]. This methodology is a lot better for measuring small earthquakes (ML<3.0) with low SNRs compared to traditional methods. Figure 4, illustrates the prediction results, as well as the errors are locating the sources of the earthquakes. To train PhaseLink network, in a study, millions of synthetic sequences were created[20]. Seismological records from the same source into one set are involved classified by associated seismic phases. Zhang and Curtis (2020) utilized inference based on variation to perform earthquake tomography[21]. Using this method, both mean and variance are obtained as outputs. The connection between a strong earthquake and a post-seismic deformation was analyzed by Yamaga and Mitsui (2019)[22]. A small data set was obtained along the Global Navigation Satellite System (GNSS): with 153 training points and 38 testing points. An RNN was employed to investigate the corresponding relations, providing much more accurate results than traditional regression methods would have provided. Meanwhile, Rouet-Leduc et al. (2017) used a random forest model in order to predict earthquakes in a laboratory setting[23]. They used acoustic signal as inputs, and attempted to predict laboratory fault failures. By using this machine learning method, they were able to correctly identify a signal from the fault zone as what it was rather than mistake it for low-amplitude noise[4].

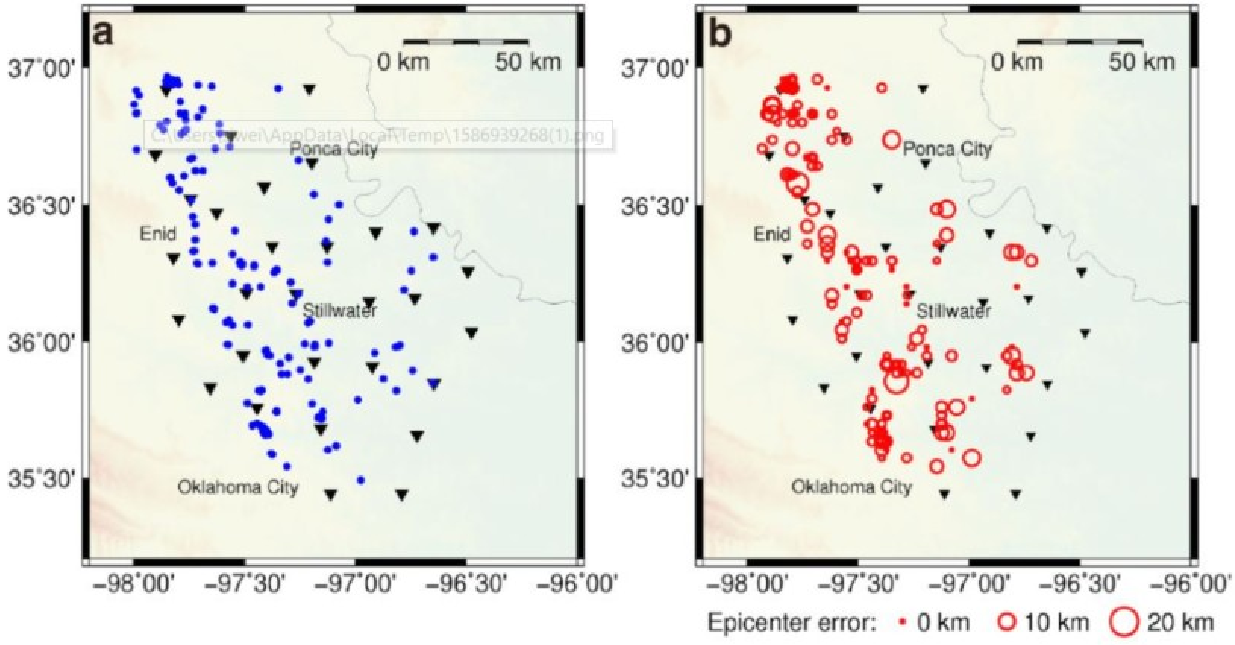


Figure 4 DL in locating earthquake sources. Black triangles represent stations. On the left, with blue dots, are shown the actual locations. On the right, with the red circles, are shown the predicted locations. The predicted epicenter error is represented by the radius of a circle[4].

Additional uses include relationship analysis among a strong earthquake and post-seismic deformation[22], as well as correlating seismic phases, that includes bundling the phase picks on manifold stations connected to a single occurence[20].

PickNet has been suggested by Wang et al. (2019) to elect natural seismic arrivals in accordance with a deep residual network[24]. The chosen arrivals were employed for seismic tomography, of Japan's subterranean structure which was reconstructed. For arrival picking and polarity classification a CNN was trained by Ross et al. (2018)[8] using a training set of 19.4 million seismograms. They managed to get exceptionally high picking and classification accurecies as or even more accurate than what would have been obtained even by highly trained professionals. Furthermore, Zhao et al. (2019) suggested taking advantage of using a U-Net for P- and S-phase arrival picking to provide much better results than the STA and LTA methods would[6][7]. Finally, Zhou et al. (2019) developed a hybrid event detection and phase-picking algorithm with CNNs and RNNs[7].

2 Machine Learning in Geophysics

Machine learning, the last few years, is taking greater and greater part as a significant multidisciplinary tool which has shown improvement in various domains of science, such as biology[25], chemistry[26], medicine[27], and pharmacology[28]. ML is an area of computer science concerned with the development of algorithms that relies on the collection of examples of the phenomenon of interest. These examples can be occurred naturally, be artificial or man-made, be generated by another algorithm. ML is in general considered to be a subdomain of artificial intelligence (AI), the concept of which was first imported by Turing[29]. Instead of being specifically programmed, a machine learning system is trained. It is given a great amount of cases that are pertinent to a task, and by identifying statistical patterns in these examples, it can develop rules to automate the task[30]. The main types of machine learning are supervised, unsupervised, semi-supervised and reinforcement[31].

2.1 Machine Learning applications

ML has strong roots in applied statistics, constructing computational models that use inference and pattern recognition rather than explicit sets of rules. It is a method based on the given data and trains a model of regression or classification via a complex non linear system with adjustable parameters based on the training set. A combination of demands defines an ML model. Consequently, in ML are applied plenty of mathematical and statistical methods and concepts, considering Bayes' rule[32], least-squares[33], and Markov models[34][35].

Figure 5 introduces a model-driven approach as well as a data-driven approach[3].

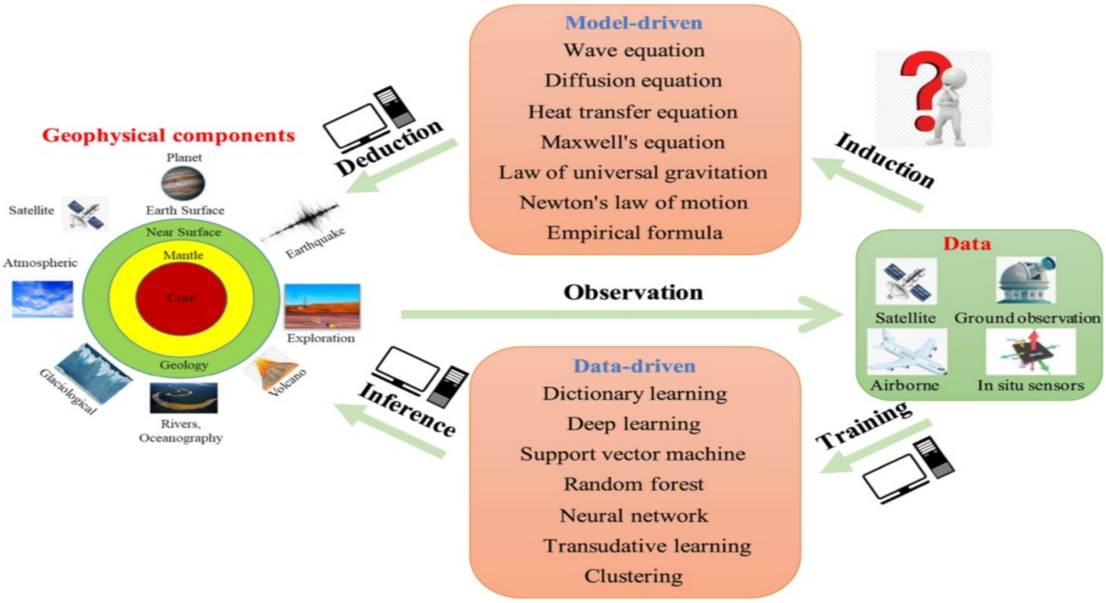


Figure 5 A presentation of a data-driven and model-driven approach. The geophysics research areas, which span through the Earth's core beyond outer space, are shown on the left. The current observational methods are shown on the right. Instances of model-driven both data-driven approaches are shown in the middle. The principles underlying geophysical phenomena are inferred using model-driven approaches from a big number of observable data that relies on physical causation, and the models are then utilized to extrapolate past or contemporary geophysical occurrences. Without taking physical causality into account, the computer first introduces either a regression either classification model in data-driven approaches. Then, using the incoming datasets, this model will carry out tasks like categorization.[3].

ML is the science area which uses algorithms to teach machines to make calculations and predictions as well as to take decisions, depending on their training. It is the way to process big data that cannot be observed, to train a model and make predictions. There are different methods to train the model, based on the wanted application. In supervised learning, the model has a trainer who already knows the results of the incoming data, can correct the free parameters while training the model, thus optimizing the model. On the other hand there is the unsupervised learning, which is a competitive learning method, as the neural networks during the whole process of the training compete each other to see which one will bring up the best results. The main purpose of this method is to create patterns based on the incoming data, instead of teaching it the relationships between incoming and outgoing data. Unsupervised learning is mostly used in clustering problems. Finally, there is the reinforcement method, which has constant contact with its environment and does not require a supervisor or trainer. This method takes feedback from the environment during the training to learn either it is performing correctly or not. It then uses this feedback to correct itself. Reinforcement is mainly used on regression and dynamic problems[36][37][38].

ML methods are used in different geophysical applications[45][46][47][48][49][50], such as earthquake detections[39], aftershock pattern analysis[40], and earth system analysis[41].

As Bergen and colleagues (2019) state in their review of ML in geoscience, the key role of ML is to achieve a quick understanding of the intricately interconnected and multifaceted processes that shape the Earth's behavior[42].

2.2 Machine Learning Tools

The 2000s introduces a change in the existing tools up to that time, which actively contributes to the later increase in adoption and research, of shallow and deep machine learning research.

Private software such as MATLAB® with the Neural Networks Toolbox and Wolfram Mathematica, or independent university projects such as the Stuttgart Neural Network Simulator (SNNS), have been the primitive machine learning software. In general those tools were closed source and difficult or impossible to expand and could be hard to run because of restricted accompanying documents. Early open-source projects contain WEKA[43], a graphical user juncture to create ML and data mining tasks. Soon thereafter, LibSVM has been made available as complimentary open-source software (FOSS), which materializes support-vector machines effectively. It is still in use today in plenty other libraries, among others WEKA[44]. Torch is an ML library, published in 2002, with an emphasis on neural networks. PyTorch, a reapplication in the programming language Python, is one of the leading DL contexts at the moment of writing[45], although its original realization in the programming language Lua has been extincted[46]. The Python libraries Theano and scikit-learn were released in 2007, flagrantly licensed[47][48]. Theano is a neural network library, a tool that was established at the Montreal Institute for Learning Algorithms (MILA) and stopped being developed in 2017 as a result of the availability of open-source deep learning frameworks by powerful industrial developers. Scikit-learn realizes various ML algorithms, involving SVMs, Random forests, and single-layer neural networks, in addition to utility functions such as cross validation, stratification, metrics, and train-test separation that are essential for the creation and assessment of strong ML models. By establishing a consolidated application programming interface, scikit-learn has formed the current ML software package (API).

3 Deep Learning in Geophysics

Deep learning (DL) is the multilayer computation on the models that classical ML cannot bring a result. It forms the -complex- neural networks[38][49]. There are three ways to develop a model, MLP, CNN and RNN. Multilayer Perceptron (MLP) is suitable for table data and it works with panda library. Convolutional Neural Network (CNN) is made for importing images. Recurrent Neural Network (RNN) is made to work with sequential data[50]. Additionally, DL aids the appraisal of earthquake locations as well as magnitudes based on signals from one individual station[51][52].

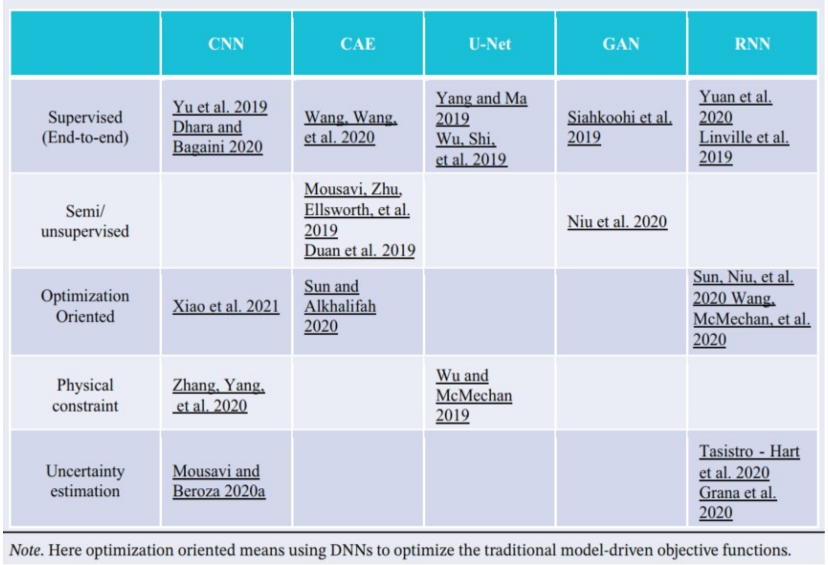
3.1 Deep Learning applications

DL lately has taken the attention of the geophysical community, due to its increasing possibilities, as also the fact that it can predict with high accuracy the complex situations, in particular the method that has found widespread application is DNNs. With DL we get rid of the well known "curse of dimensionality" as it's called. It mainly applies on geophysical research, as well on earthquakes and remote sensing. In addition, it can apply in atmospheric and space science, involving the earth structure and the water sources. The focus of the discussion primarily revolves around the future advancements in DL for unsupervised learning, transfer learning, multimodal DL, federated learning, uncertainty estimation, and active learning[3].

As a powerful artificial intelligence method, DL is expected to uncover geophysical resources and pass along expert knowledge using computer-aided mathematical algorithms. The use of DL as a tool for the vast majority of minutes geophysics is still in its babyhood, despite the fact that it has been successful in several geophysical applications, like seismic detectors or pickers. The principal issues comprise weak nonlinearity, low signal-to-noise, and insufficient training data. The absence of training samples for geophysical implementations compared to other industrial sectors is one of these obstacles. Several advanced DL techniques, like semi-supervised and unsupervised learning, transfer learning, multi-modal DL, federation learning, and active learning, have been suggested to solve this issue[3].

DL's milestone accomplishment came after 2015, like VGGNet[53], ResNet[54], AlexNet[55], and AlphaGo in 2016. Because remote sensing is a common technique extensively employed in several fields, the initial admission of DL in geophysics-related topics in 2016 and 2017[56][57][58][59] focused on remote sensing. Further geophysical fields, such as exploration geophysics[60] and earthquake studies[61], began using DL in 2018 and 2019.

Complex networks like CNN, RNN, and GAN models were developed from plain FCNN methods. Regarding the training set, whilst newly work has begun to consider unsupervised learning[62] and combining DL with a physical model[63][64], early works employed computer vision-based end-to-end training, where necessitates a lot of annotated labels. More tasks employing DL techniques in 2020 concentrated on uncertainty[51][65][66]. Further examples are given in Table 1. These tendencies indicate that more and more researchers attempt to develop DL methods particularly directed for geophysical jobs in order to make the methods very usable.



	CNN	CAE	U-Net	GAN	RNN
Supervised (End-to-end)	Yu et al. 2019 Dhara and Bagaini 2020	Wang, Wang, et al. 2020	Yang and Ma 2019 Wu, Shi, et al. 2019	Siahkoohi et al. 2019	Yuan et al. 2020 Linville et al. 2019
Semi/ unsupervised		Mousavi, Zhu, Ellsworth, et al. 2019 Duan et al. 2019		Niu et al. 2020	
Optimization Oriented	Xiao et al. 2021	Sun and Alkhalifah 2020			Sun, Niu, et al. 2020 Wang, McMechan, et al. 2020
Physical constraint	Zhang, Yang, et al. 2020		Wu and McMechan 2019		
Uncertainty estimation	Mousavi and Beroza 2020a				Tasistro - Hart et al. 2020 Grana et al. 2020

Note. Here optimization oriented means using DNNs to optimize the traditional model-driven objective functions.

Table 1: Representative Instances of Literature that Extends Further than End-to-End Training and Uses Alternative Network Architectures[3].

3.1.1 Methods in Deep Learning

A main problem nowadays is the huge quantity of data that need to get processed, while at the same time we want a high quality analysis of the earth[3].

In figure 6 is presented the published papers that concern to AI in two important geophysical associations, which are the Society of Exploration Geophysics (SEG) and the American Geophysical Union (AGU). It is obvious that the application of DL approaches resulted in an exponential growth in both libraries. Additionally, DL has provided plenty amazing outcomes to the geophysical community. A remarkable example, is on the Stanford Earthquake Data set (STEAD), the accuracy of earthquake detection is improved to 100% in contrast to 91% that is provided by the traditional STA/LTA method[10][13]. DL makes it feasible to characterize the land with high analysis on a large scale[63][67][68]. At last, DL can also be used to discover physical concepts[69], such as the fact that the solar system is heliocentric.

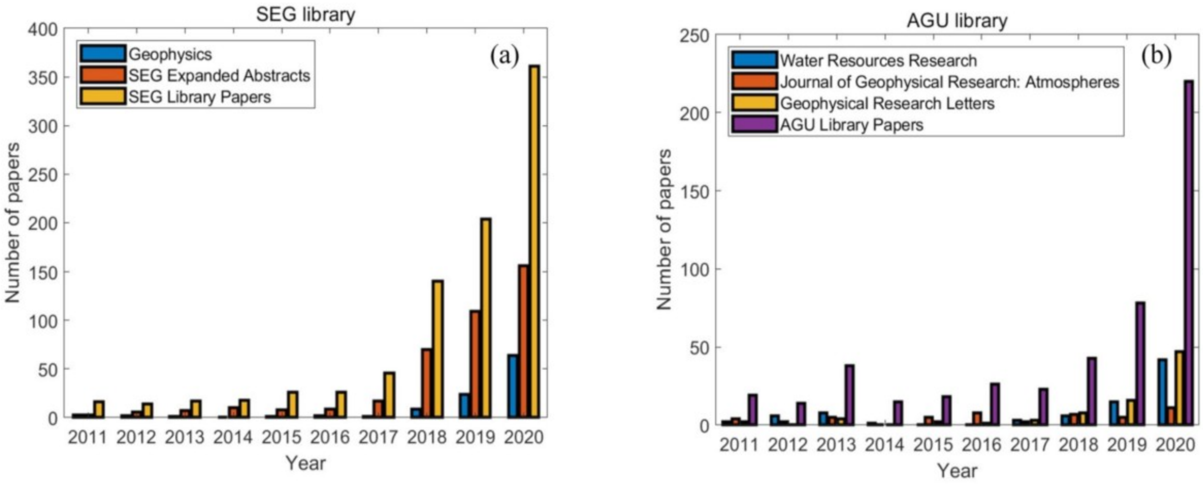
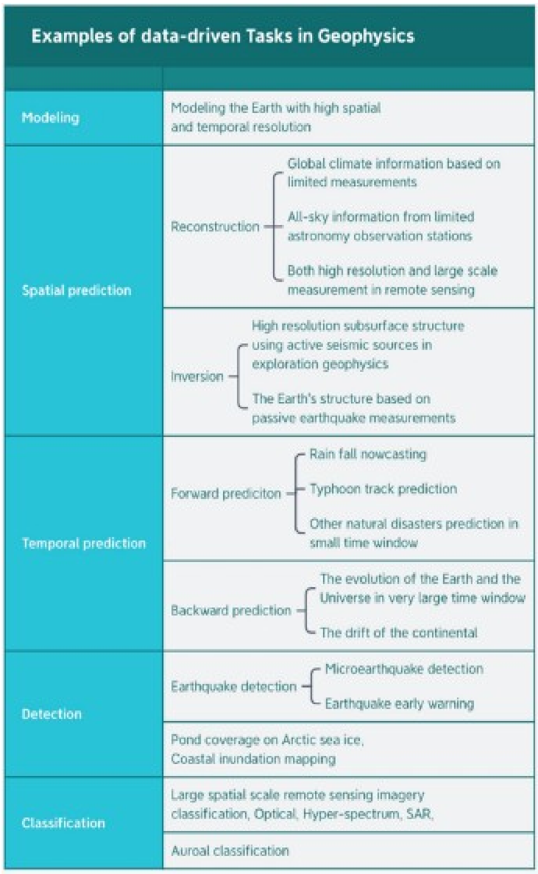


Figure 6 Both (a) and (b) are statics papers that are associated to artificial intelligence (AI) that may be found in the SEG and AGU libraries, respectively. Geophysics refers to the SEG's premier journal in (a). SEG Expanded Abstracts of the SEG annual meeting are referred to as Expanded Abstracts. The papers created within the SEG digital repository are referred to as SEG Library papers. The names of AGU's top journals are represented in the first three heading in the legend, in (b). The AGU digital library papers are represented by the fourth caption in the legend[3].



Examples of data-driven Tasks in Geophysics		
Modeling	Modeling the Earth with high spatial and temporal resolution	
Spatial prediction	Reconstruction	Global climate information based on limited measurements All-sky information from limited astronomy observation stations Both high resolution and large scale measurement in remote sensing
	Inversion	High resolution subsurface structure using active seismic sources in exploration geophysics The Earth's structure based on passive earthquake measurements
Temporal prediction	Forward prediciton	Rain fall nowcasting Typhoon track prediction Other natural disasters prediction in small time window
	Backward prediction	The evolution of the Earth and the Universe in very large time window The drift of the continental
Detection	Earthquake detection	Microearthquake detection Earthquake early warning
	Pond coverage on Arctic sea ice, Coastal inundation mapping	
Classification	Large spatial scale remote sensing imagery classification, Optical, Hyper-spectrum, SAR,	
	Auroal classification	

Table 2: Examples of Data-Driven Tasks in Geophysics[3].

3.2 Deep Neural Network Architectures

A regression or classification model that is made up of layers of neurons that are comparable to a human's brain is called an Artificial Neural Network (ANN). If the ANN now has more than one layer, this creates a Deep Neural Network (DNN), which gives as result the known as called Deep Learning. DL is at the core of a new ML method development[70]. There are two main ways that a DL works, supervised and unsupervised, reliant on whether tags are available or not. For regression and classification tasks, the unsupervised approach trains the DNN by pairing the input and the labels, while for clustering and pattern recognition the unsupervised approach is more preferable as it updates the parameters by creating a compact inner representation. Furthermore, DL includes as well semi-supervised learning in which part of tags are available and reinforcement learning in which an environment designed feedback for DNN is provided by human-designed environment. Figure 7 briefly illustrates the relationship of AI to DL as also the classification of DL approximates. DL's performance even outperforms the human brain in certain works, like image classification (5.1% vs. 3.57% for top-5 classification errors, He et al.,2016) as well as the game GO[3].

The training set, the network architectures and, the parameters optimization are the basic components of DL. DNN architectures differ depending on the application; In this section, are presented some of the most commonly used architectures.

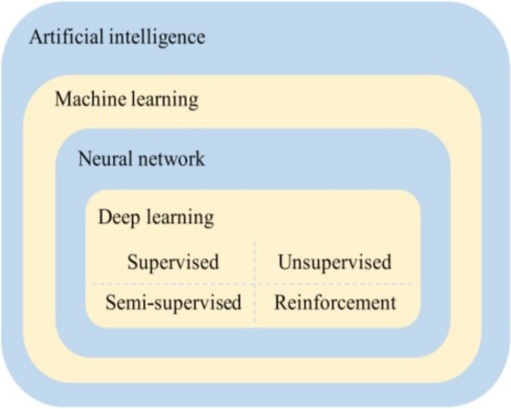


Figure 7 AI, ML, neural networks, and DL confinement relationships as well as the classification of deep learning methods[3].

3.2.1 Neural Networks

The very first neural network machine was built by Minsky and shortly after it was followed by "Perceptron", a binary ruling barrier learner[71]. This decision has been calculated as below:

$$\begin{aligned} o_j &= \sigma\left(\textstyle\sum_j w_{ij}x_i + b\right) \\ &= \sigma\left(a_j\right) \\ &= \begin{cases} 1 & a_j > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \qquad (1)$$

It is describing a linear system where o is the output, a the linear activation of the input data c, i the index of the source and j target node, w the training weights, b the trainable bias and σ a binary activation function as it is illustrated in figure 8.

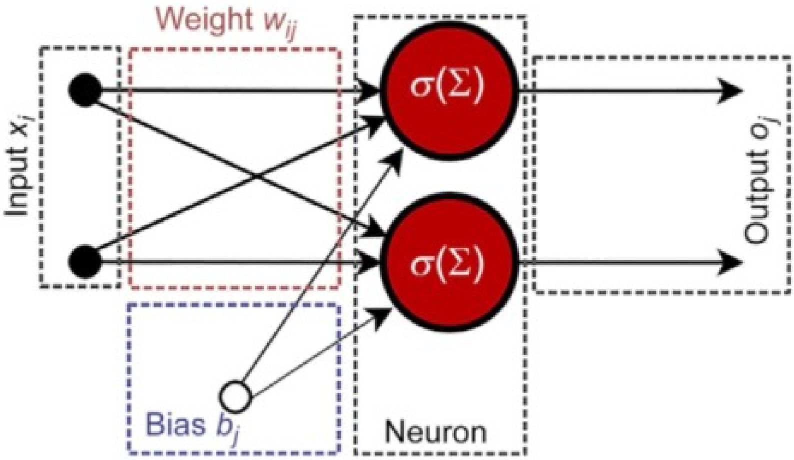


Figure 8: A single layer neural network as it is outlined in equation (1). 2 xi inputs are multiplied by the wij weights and summarized with the bj bias. Finally, an activation funcition σ is implemented to get out the oj outputs[84].

Particularly the activation function σ, since its inception, it has obtained great attention. A binary σ, during this time, turned into extraordinary and was substituted by nonlinear mathematical functions. Neural networks are usually trained by slope descent, thus, differentiable functions such as sigmoid or tanh, enabling the activation of any neuron in a neural network to be continuous as it is presented in figure 10. This notion is an extension by DL[72]. It involves combining multiple layers of neurons to form a neural network as it's shown in figure 9.

These deep networks learn performances with manifold levels of ablation, and can be expressed in conditions of the up coming level by using equation (1)

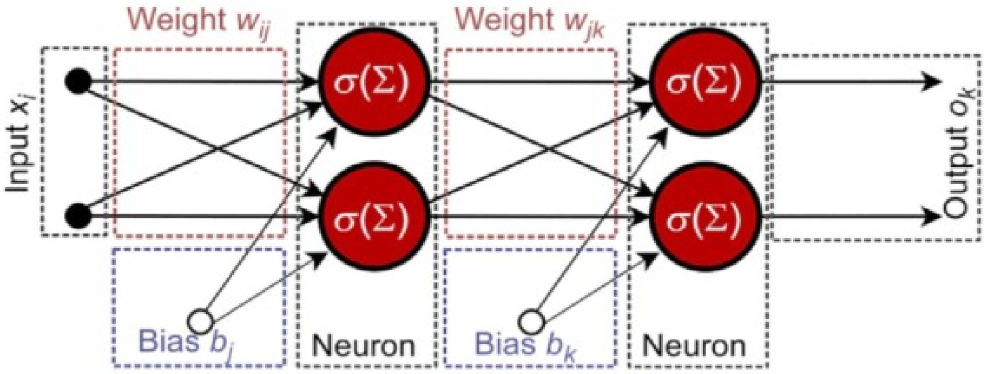


Figure 9 Deep multilayer neural network as it is outlined in equation (2)[84].

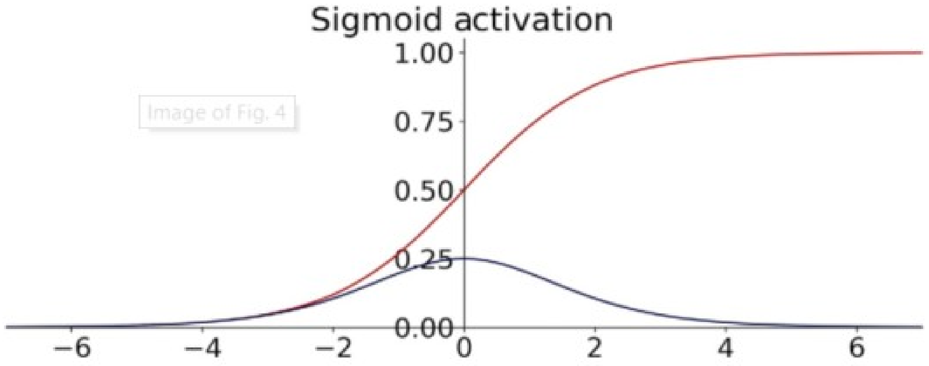


Figure 10 The red line stands for the sigmoid activation function and the blue line stands for derivate to train a multilayer neural network as it is outlined in equation (2)[84].

$$\begin{aligned} o_k &= \sigma\left(\sum_k w_{jk} \cdot o_j + b\right) \\ &= \sigma\left(\sum_k w_{jk} \cdot \sigma\left(\sum_j w_{ij} x_i + b\right) + b\right) \end{aligned} \qquad (2)$$

These structural elements of multilayer sigmoid activation neural networks are applied to seismic inversion by Roth and Tarantola[73]. With small training data, they achieve with success to invert low-noise and noise-free data. The authors point out that the approach is prone to error when the signal-to-noise ratio is low and when the noise sources are coherent. Other applications involve electromagnetic subsurface localization[74], magnetotelluric inversion via Hopfield neural networks[75], and geomechanical micro-fractures modeling in triaxial compression tests[76](Figure 10).

3.2.2 Fully Connected Network

As it is shown in Figure 11a, a fully connected network (FCNN) is an ANN which is consisted of fully linked layers, wherein one layer's inputs connect to each unit in the later layer. The weighting sum of the entries traverses a non-linear activation function f on a unity. The traditional DL f function, consisting of the sigmoid, tanh, and Rectified Linear Unit (ReLU) functions, is shown in figure 12a. The adaption and generalization capabilities of the model, are importantly affected by the amount of layers in a FCNN. Though, the reasons that FCNNs were limitated in few layers, is the calculation ability of hardware's availability, the vanishing both explosion gradient issue that can be occured while the optimization is on, etc. ANNs are beginning to turn more deeply as the hardware both optimization algorithms advance.

On the other side, in case that raw data set is the direct input of the FCNN, it will be needed huge parameters, as each features is matched with each pixel, notably for high dimensional inputs. Features are essentially utilized to minimize the input layer's size, which reduces the number of parameters in the model. FCNN completely disregards the input's structural organization and relies solely on preselected features. Automated feature selection algorithms are suggested[77], however they demand a lot of computing resources. CNNs (Figure 11b) were recommended to allocate network’s parameters along convolutional filters, to minimize the amount of parameters in an FCNN as well to be taken into account the local consistency in an image.

3.2.3 Deep Neural Networks

For a decade, CNNs have been quickly developed for image partition as well as classification, led by VGGNet[53] and AlexNet[55]. They are also used in image denoising[78] and super-resolution tasks[79]. Instead of using selected features as an input set, a CNN uses original data, as well it uses convolutional filters to limit the inputs of a neural network to a local range. The convolutional filters are shared by various neurons in the same layer. Figure 12b illustrates, one convolutional layer, one nonlinear layer, one batch normalization layer, and one pooling layer make up a typical CNN block. The main elements who consist the CNN are convolutional layers and nonlinear layers. The deterrence of gradient explosion and stabilize the training happens by the batch normalization layers. To extract principal features, pooling layers subsamples the input. The plainest CNNs are called as vanilla CNNs, that are CNNs with plain consecutive structures (like those that appear in a vanilla FCNN). Vanilla CNNs are trusted for the majority of the employments in geophysics, like denoising, interpolation, velocity modeling, and data interpretation, if there are available plenty training samples and labels. CNN is invariant by small changes in inputs due to pooling layers. The loss of information caused by pooling layers prevents CNN from characterizing the changes in the input. It is suggested to use capsule networks[80] to preserve invariance and characterize changes simultaneously. To be this accomplished, vectors are used in place of scalars as the inputs and outputs for the neurons. The likelihood that one entity existing is represented by the length of the vector. The vector's orientation represents the entity's parameters.

Further DL network architectures based on vanilla FCNNs or CNNs have been suggested for specialized tasks. With the use of an encoder and a decoder, an autoencoder learns to rebuild the inputs with meaningful representations[81]. To map inputs to a latent space, the encoder uses non-linear layers. The decoder decodes the latent features into the original data space using nonlinear layers. Autoencoders are trained in a self-supervised manner. Imposing additional constraints on the network achieves the essential representation. For instance, undercomplete autoencoders restrict the latent space's size to be less than that of the inputs, allowing the encoder to extract critical features. The latent space is typically subjected to a sparse regularization by sparse autoencoders, which are typically overcomplete with a bigger latent space than the input space. By making the autoencoder sturdy to the input's fluctuations, denoising autoencoders or contractive autoencoders develop useful representations. Convolutional autoencoders, Figure 11c CAE, used in the encoder convolutional layers and in the decoder deconvolutional layers.

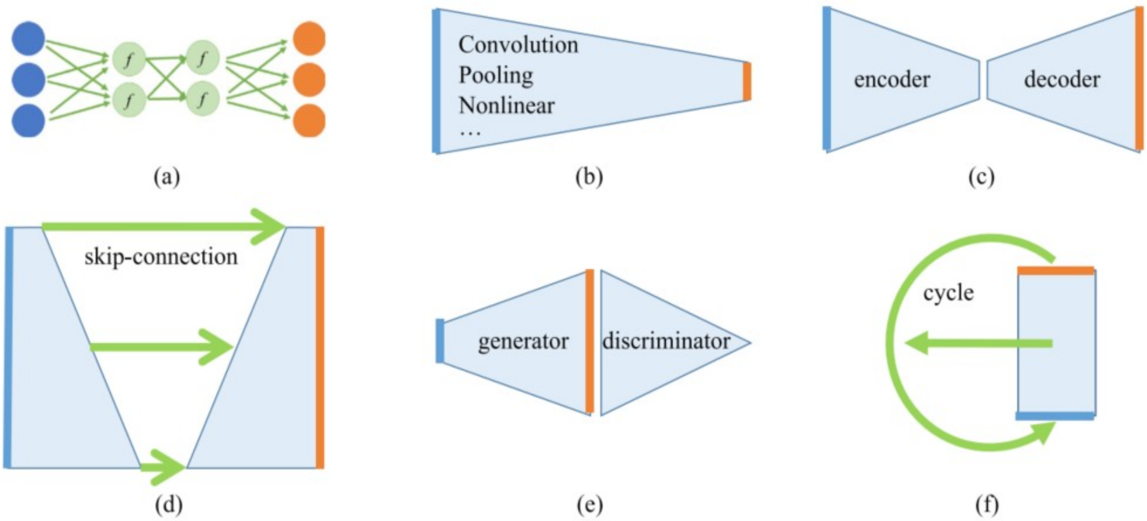


Figure 11 Deep neural network (DNN) sketches. The orange lines denote outputs, whereas the blue lines represent inputs. The data dimension is represented by the length of both the orange and blue lines. Intermedia connections are indicated by the green lines. (a) Each unit on the layer after the first is connected to the inputs of the layer before it in a fully connected neural network (FCNN).. A nonlinear function of activation is denoted by the letter f. It is eliminated the layer specifics in (b-f) while maintaining the overall form of every network architecture. (b) Convolutional layers, pooling layers, nonlinear layers, and other layers are cascaded in a vanilla convolutional neural network (CNN). Depending on the convolutional strides used in CNN, the outputs of the convolutional layers can be equal to or less than the input. The size that the extracted characteristics have will be reduced by pooling layers. In either classification or regression tasks, the output frequently has the same either a lesser dimension in comparison to the input ((b) illustrates the latest situation). The discrepancy among both classification and regression employment lies in the outputs where in regression are variables that are continuous, whilst the latter are discrete variables that represent categorization. The dimension of the lurking attribute space at CAE is able to be greater or smaller compared to the dimension of the data space, (c) presents the latest situation. (d) In U-Net, skip connections are utilized to elevate low-level characteristics to a higher level. (e) A GAN uses lowdimensional random vectors to create a sample derived from the generator, which is subsequently rated either as true either as false from the discriminator. (f) The concealed state or output of an RNN is utilized as input over a cycle.[3].

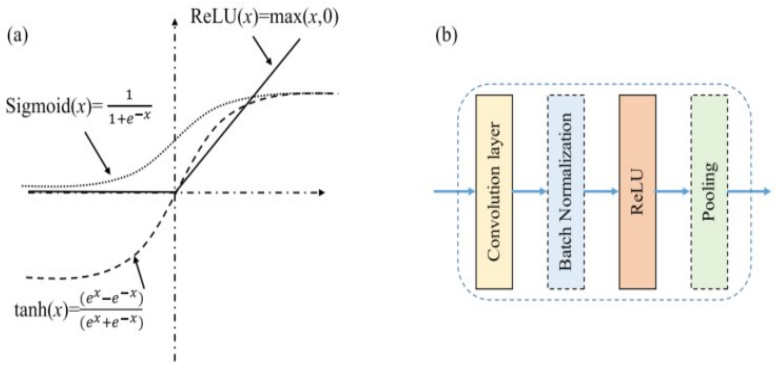


Figure 12 Deep neural network (DNN) architectures' details. (a) Nonlinear layer activation functions. ReLU is often used because its gradient is simple to compute and can prevent gradient vanishing. (b) Illustration of a standard barriering convolutional neural network (CNN). The essential components of one CNN block are the convolutional layer and the ReLU layer (nonlinear layer). Gradient explosion can be avoided with the batch normalizing layer. By subsampling the input, the pooling layer can extract features[84].

3.2.4 Recurrent Neural Network

U-Nets[82], as shown in figure 11d, have U-shaped structures and skip connections. Low-level characteristics are brought to high levels by skip connections. U-Net had initially been suggested as a method for image departmentalization, but nowadays it has also been implemented for earthquake data analysis, inverting, and interpreting. Because of the U-shape architecture of a shrinking aisle as well as an extended aisle, every point of data on the output encompasses the whole information that was provided by the input, rendering the approach useful for photographing data in a variety of domains, such as inverting speed from seismic records. For a trained U-Net, it is necessary for the input quantity in the test set to match that of the training set. In case that, the size does not line up with the demand for U-Net, the data have the necessity to be processed patch-wisely.

In adversarial training, a GAN, figure 11e, can be used with one generator to generate a fake image or whatever different type of data and one discriminator to discriminate the generated ones from the real ones. The real data set and the generated data set are mapped to labels one and zero, while training the discriminator. In addition, all datasets are mapped to the label one, once the generator is trained. Such a game at last authorizes the network of generation to produce bogus images where the discerning network is incapacitated to discriminate between the real images. To create samples with distributions like those of the training set, a GAN is utilized. The samples produced are employed to expand the training set or simulate actual situations. CycleGAN is actually a GAN that has been extended; in addition, it's been suggested to feature a pair of generators and a pair of distinguishers to process signals.[83]. At CycleGAN, bi-directional cartography is trained to match a pair of datasets one into the other. Because CycleGAN's training set is not required to be paired like a vanilla CNN, it is quite simple to build training sets for geophysical applications.

RNNs constitute the last sort of structure employed by geoscience.[84]. RNNs, figure 11f, are frequently utilized for tasks linked succedent data, wherein the history of the inputs that are feeding the neural network is engaging the present state of the network as a result of being dependent on them. Long short-term memory (LSTM)[85] is a widely used RNN that evaluates the amount of historical information that is remembered or lost. In comparison to the vanilla RNN, which has a vanishing gradient problem for long sequences, the principal advantage of LSTM is its ability to handle longer time periods of data. Consequently, the LSTM's inference accuracy improves as more historical information is taken into account. Gated recurrent unit (GRU)[86] is an alternation of LSTM with a straightforward architecture. In contrast to LSTM, GRU has analogous performance with less parameters, so that is computationally low-cost. The use of RNN in geophysical applications is primarily concerned for predicting the next sample of a temporally or spatially sequential data set. RNNs are also used to model seismic wavefields or signals by simulating the time-dependent discrete partial differential equation[3].

In order to take historical leverage into account, in RNNs, the input of one network arises from the output of the previous process in time-sequenced data processing applications. RNNs are used to predict new outputs from a consecutive input, e.g. to be able to predict the next word in an input sentence. The amount of historical information which get into account, rises the prediction precision of LSTM. RNNs are used to predict the upcoming sample of a time-sequenced or spatially-sequenced data set in geophysical applications. Wavefield simulation based on a time-dependent network shape are also performed using RNNs[4].

LSTMs are able to resolve common problems by applying particular gates which normalize flow in an LSTM cell, i.e., input gate, forget gate, and output gate, illustrated in figure 13. The internal cell receives input values from the input gate. The forget gate replaces the existing state with a new one. Lastly, the output gate controls the direct contribution of the input value to the output value, as well as the cell's internal state. Furthermore, a peephole feature aids in training by acting as a shortcut among inputs and gates.

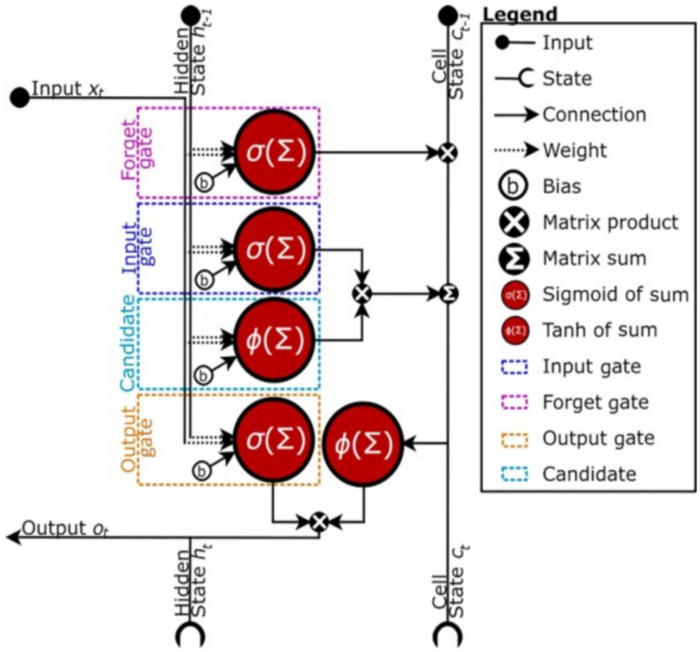


Figure 13 LSTM architecture schematic. The hidden state both cell state are processed with the input data. By implementing an input, forget, and output gate, the LSTM avoids the exploding gradient issue.[84].

A lot more particular architectures exist and are being developed which offer several benefits. For instance are Siamese networks for single shot image analysis[87], transformer networks which have greatly substituted LSTM and GRU in language modeling[88], or attention in DNNs as a general mechanism[89].

In the field of geosciences, neural network architectures have been changed and implemented to various problems. Some data types found in each geoscience discipline, make each architecture type especially well-suited. Yet, the adoption of ML tools and implementations have speeded up in areas where data exists in a machine-readable format. For instance, Ross, Meier, and Hauksson (2018) successfully implemented CNN for seismic phase detection and relied on a expanded catalog of hand-selected data[90], thus generalizing the work[91]. It is important to note that synthetic data, or data that is especially sampled, can import an implied bias into the network[92][93]. Yet, the blackbox quality of ML models, in particular, makes them adaptable and strong tools that have been used in every subsector of geosciences[84].

3.3 The Theory behind Deep Learning

Physical model-driven procedures generate an optimization objective loss function with a further restriction. Given a large training set, a pairing among x and y is created by training in data-driven routines, that is particularly useful in circumstances when L is not exactly known.

Italic letters present scalars, bold lower case letters show vectors and bold upper case letters, matrices. In geophysics, a large amount of regression or classification tasks can be decreased to

$$\mathbf{y}=\mathbf{L}\mathbf{x} \qquad (3)$$

where **x** represents the unknown parameters, **y** represents the partial known observation and **L** is an onward or depraved operator in the observation of geophysical data, like contamination or subsampling, or physical response. Though, it’s common **L** to be ill-condioned or not invertible, or even unknown. The inverse of **L** is pricipally accomplished by two ways: physical model-driven and data driven.

Geophysical issues are handled by DL as classification or regression issues. Using a DNN F, we can approximate x from y,

$$\mathbf{x} = F(\mathbf{y};\boldsymbol{\Theta}) \qquad (4)$$

where **Θ** denotes the DNN parameter set. **x** is a one-hot encoded vector that represents the categories in classification tasks. **Θ** is received by constructing a high-dimension approximation among two sets **X**={**x**$_i$, i=1...N} and **Y**={**y**$_i$, i=1...N},i.e. the labels and inputs. The approach is accomplished by minimizing the subsequent loss function to attain an optimized **Θ**:

$$E(\boldsymbol{\Theta};\mathbf{X},\mathbf{Y})=\sum_{i=1}^{N}\|x_i-F(y_i;\Theta)\|_2^2 \qquad (5)$$

An approach based on the gradient is possible to be employed to enhance **Θ**, only if F is the differential. Yet, while the $\nabla_\Theta E$ is calculated, a large Jacobi matrix is included, something that makes it unattainable for large-scale datasets. To avoid the calculation of the Jacobian matrix and to calculate $\nabla_\Theta E$, a back-propagation method is recommended[94]. In unsupervised learning, the label **x** is unknown, necessitating the use of extra restrictions, such as making **x** and **y** identical.

With enough hidden neurons, any complex function can be represented using an ANN having one hidden layer as well with nonlinear operators. There is the necessity to be an extensive training set for the adaption of an ANN with plenty hidden neurons. In figure 14 is presented the hierarchical structures of filter in DL.

On the one hand, DL can be thought of as an ultra-high dimensional nonlinear mapping from data space to feature space or target space, where the nonlinear mapping is represented by a DNN. As a result, DL is fundamentally a high-dimensional nonlinear optimization issue. On the other hand, RNNs are essentially a solution of the usual differential equation by the Euler method[95]. Since the major goals of a generative adversarial network[96][97] (GAN) are manifold learning and probability allotment transformation, which is, transformation among the provided white noise and the data allotment[98], the theory of optimal transportation can also be construed by a GAN. RNNs and GANs are two particular DNNs.

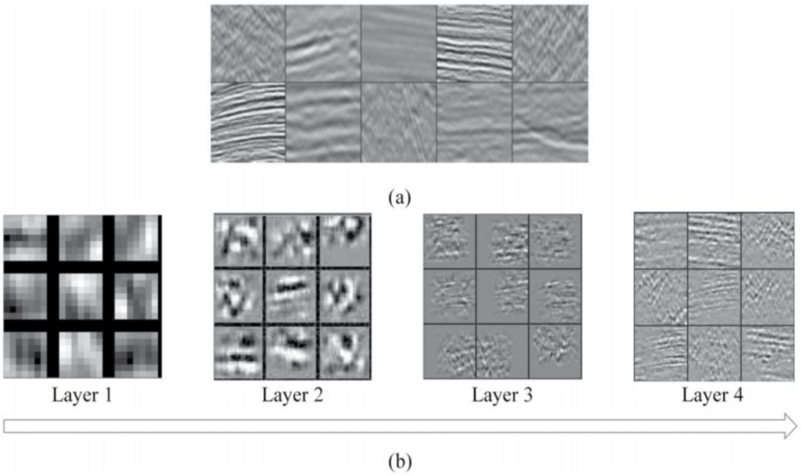


Figure 14 The DL learned characteristics. (a) Samples of training. (b) Nine of the learnt filters are displayed in each layer. In various strata, numerous hierarchical structures can be seen. Edge structures are displayed in layer 1, small structures of earthquake occurences are displayed in layer 2, and a small segments of seismic section are displayed in layer 3. The boundary impact of the convolutional filter may be the reason, layer 2 and layer 3 filters are blank at edges. Larger seismic parts are included in Layer 4, which are approximations of the training data. DNN attempts to understand the similar both hierarchical patterns that constitute the data, which makes the filters that are used by the 4th layer, to seem to be more alike to one another than the datasets used for training[3].

3.4 Deep Neural Network Applications

The last years, it has been an important issue the joint inversion of data from various sources, for improving the resolution of inversion[99]. The ability to fuse information from multiple inputs is one of the benefits of DNNs. Insertion of seismic and gravity elements is derived from various sources [100][101]. Gathering data from various sources may alleviate the issue of a limited number of training samples. Furthermore, the quality and reliability of DL methods can be improved by using multimodal data sets[68].

The most instant way to apply DL to geophysics, is to transfer geophysical works to computer vision operations like denoising or classification. Though, in some geophysical implementations, the features of geophysical workloads or data differ from those associated with computer vision. Geophysics, for instance, has high-dimensional as well as large-scale data with less-commented labels.

Typically, it is need to train a DNN for a certain dataset and a certain project. For instance, a DNN may be efficient at processing land data however inefficient at processing marinetime data, as also a DNN might be efficient at fault detection still inefficient at facies classification. For increasing the potential for reuse of a trained network across various datasets or applications, it is suggested transfer learning[102]. The well known fine tuning process, is when in transfer learning with different datasets, the optimized parameters for one data set can be used as initialization values for learning a new network with another data set. Instead of starting from the ground up to train a network using haphazardly initiated weights, fine-tuning is usually much faster and easier. In the case of cross-task transfer learning, our consideration is that the extracted characteristics should be the identical in several tasks. This decreases training time by copying first layers of a trained model into a new model for a different task. An extra advantage of transfer learning is the ability to quickly apply the learned characteristics to new tasks and datasets using a small amount of training samples. Figure 15 illustrates this pair of transfer learning approaches.. Other issues in transfer learning are how characteristics can be transferred across tasks and datasets[103][104].

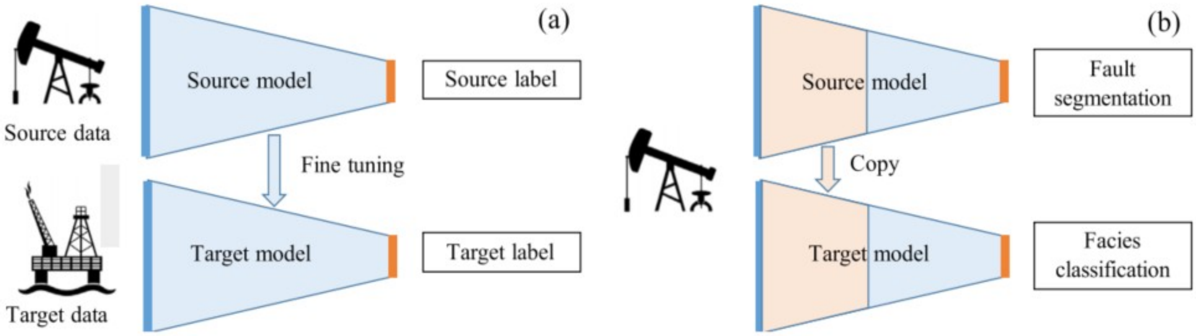


Figure 15 Transfer learning diagrams. (a) Transfer learning among several data sets. One trained model's parameters might be used as initialization conditions for another model. (b) Transfer learning to various duties. A trained model's primary layers can be replicated in a different model.[3].

4 Earthquakes

Earthquakes are ground motions under a period of time, during which Fourier shows that these motions consist of a series of frequencies, and that they bring their own phase and amplitude[105].

Seismology investigates earthquakes and elastic waves, but it is also much more than that[106]. Seismology is a very useful tool to humans. With the years this field has met a big increase in popularity. Seismometers can detect and measure massive data even for events that are happening many kilometers away from the source. This means that we can have measures and data from rough places and about events that can be dangerous and risky for people's lives, such as earthquakes, rock falls, and rock avalanches, etc in mountainous areas, and be able to analyze them[107].

4.1 Detection of Earthquake

The initial uses of ML for earthquake detection concentrated on distinguishing the spectra of seismic waveforms caused by earthquakes from those caused by explosions[108][109]. These resolutions led the way in the use of neural networks to detect characteristic seismic signatures of earthquakes. They were shortly joined by studies on the identification of earthquakes from seismic signals and their distinction from noise. Wang and Teng[110],in 1995, discovered that neural networks trained with Landers earthquake aftershock data exceeded the findings of a traditional threshold classification strategy, STA/LTA. Similar findings were found by Tiira[111] in 1999, who analyzed Finnish remote sensing data and mentioned that a neural network approach could be enhanced by 25% on the training catalog.

All the aforementioned studies used shallow forward neural networks (except Tiira[111], who used a bit more complex plain recurrent networks). They were trained with few instances using hand-built characteristics, either waveform spectra, or STA/LTA functions, or both, anywhere from a few dozen to a few hundred.

The majority of the studies listed below are based on the same instruments, as well as relatively small amounts of datasets, with examples ranging from 100 to 1000[112][113][114][115]. Nevertheless, The bulk of these investigations also embedded ML techniques other than feed-forward networks, such as support vector machines (SVMs) [112][115], random forests[114], or logistic regressions[114], which were discovered to consistently beat the basic perceptron[116] models. By using either hidden Markov models [117] or dynamic Bayesian networks[118], probabilistic graphical models (except for feed-forward networks), have been under investigation as a parallel algorithmic approach to detect seismicity. These approaches offer the additional benefit of associating a true Bayesian measure of uncertainty with each detection, which can be used to assist in the filtering out of false positives. In both studies, the authors note that there were many false detections generated by their algorithms, but the uncertainty of these false detections were well above the uncertainty associated with real seismic events. Research has also been conducted on unsupervised algorithms that rely on the similarity of regrouping event waveforms[119][120][121].

Lately, researchers have started to train models on bigger datasets[122] [123][124][125], which allows the use of more complicated neural network architectures and automatic attribution extraction. Mousavi[123] trained a residual network built on convolutional and recurrent units employing a database of 250,000 earthquakes and 250,000 noise samples in California. It is remarkable that when they attempted to extend their model from California to Arkansas, the model generated a greater proportion of false positives, implying that complicated models developed on huge databases may struggle to apply to other places.

As it is illustrated in figure 16, in total the literature analysis presents that ML algorithms in earthquake detection definitely outperforms a standard STA/LTA approach, especially because of the lower false positives rates. Less clear is the comparison with the algorithms of template matching. A few clear benefits of ML earthquake detection over template matching include (a) faster computational time (after training the model), and (b) the ability to detect events not present in the template list (depending on how well the model is generalized). ML algorithms, on the other hand, may produce more false positives than template matching. Further advancements of Bayesian deep networks trained on big datasets to connect obvious uncertainty to every detection, such as that described by Gu, Marzouk, and Toksoz[126], would be a straightforward improvement over the current state of the art.

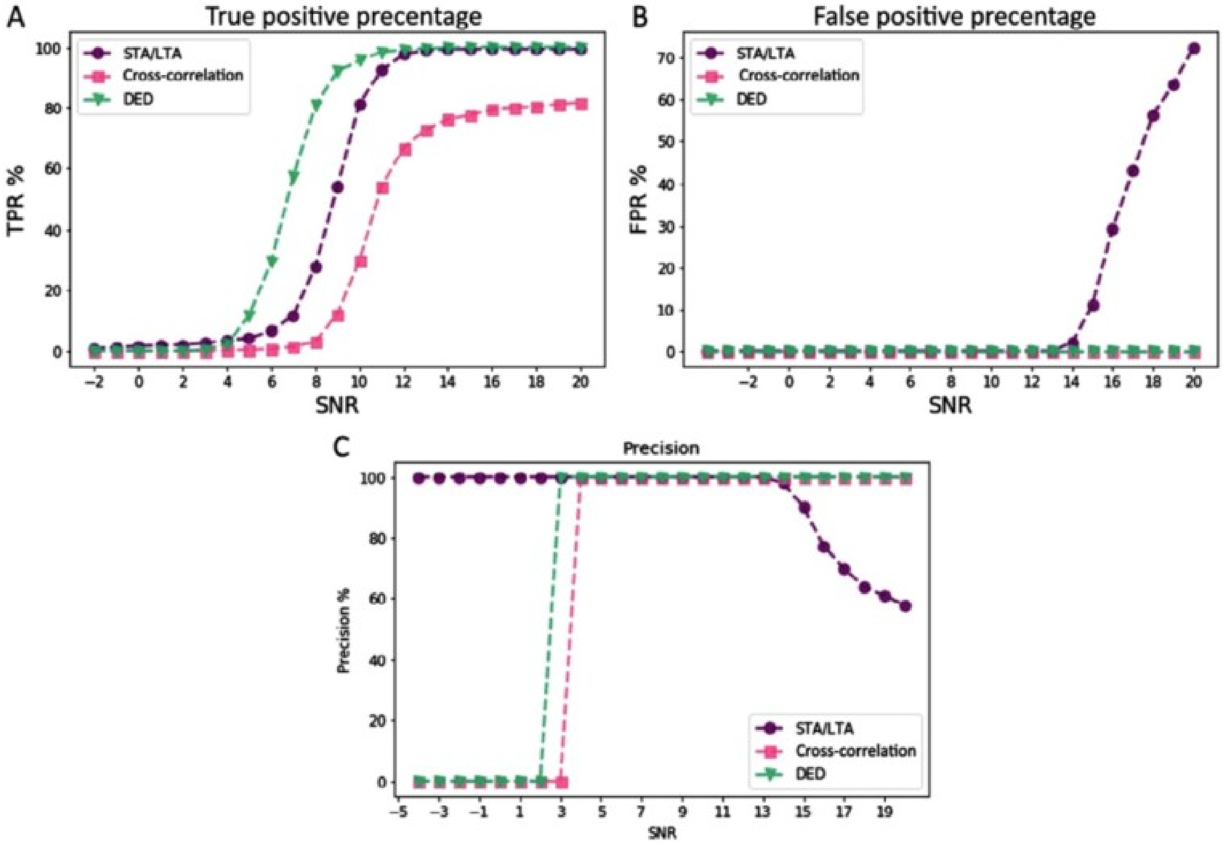


Figure 16 Comparison of crosscorrelation, STA/LTA, and neural network detection (DED) on semi-synthetic data at various signal-to-noise ratios. By Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C[13] [127].

It is especially interesting to compare template matching with CNN detection. Template matching is efficiently a single 1D convolutional layer employing a kernel of known weights for every template, since convolutional layers execute a cross-correlation functionality. On the contrary, because the pooling processes modify the resolution of the filtered input waveform across the convolutional layers of a neural network, manifold smaller templates at various analyses are learned.

Another interesting contrast is the generalization of earthquake detection methods. STA/LTA algorithms are apparently simple to generalize to different areas and/or seismic stations, but they tend to produce a significant number of false positives. Template matching algorithms, on the other hand, don't transfer well and can only detect occurrences for which templates have been identified by analysts. Because of its ability to learn templates and identify events that differ from those comprised in training, ML algorithms are promising for developing robust models that generalize. But, models that generalize well to aerias out of the training domain do not appear to be part of today's technology. This is tricky because it prevents such models from being used to analyze regions of interest with poor instrumentation or where few events have been cataloged. Enhancing the generalization of these models may be a possible domain of future research.[127]

4.2 Denoising methods

Earthquake data is infected with several kinds of noise, including random background noise, high-energy ground rolls traveling between the surface that cover-up useful signals, and manifold reflections among interfaces. Evacuating noise and ameliorating the SNR concerning signals is a long term issue in exploration geophysics. Traditional methods denoise some types of noise by parsing the respective features, using hand-crafted filters or regularization[128]. However, homemade filters fail when both noise and signals are located in the same attribute space. When used for seismic denoising, feature selection is evaded by DL methods. For instance, by learning a nonlinear regression, the U-Net-based Deep Denoiser can distinguish signals from noise[129]. Furthermore, with DnCNN[78], also known as CNN for denoising, the exact same design may be used for a trio of kinds of earthquake noise. It accomplishes a high SNR[130] as long as a counterpart training set is manufactured. But the journey is far from over. The capacity of a DNN to generalize to field data is poor when trained on synthetic data. Transfer learning[102] can be used to denoise the field data to make the network reusable. Sometimes it is hard to receive the labels of the clean data, so a feasible option is the utilization of manifold tests with white noise created by users to resemble actual white noise[131].

Figure 17 presents an example of scattered ground roll attenuation[130]. Scattered ground roll is primarily noticed in desert regions and is due to the scattering of ground roll caused by lateral variations in the near-surface properties. Scattering is hard to eliminate due to occupancy of the frequency domain as reflected waveforms. Scattered ground roll was successfully removed using DnCNN.

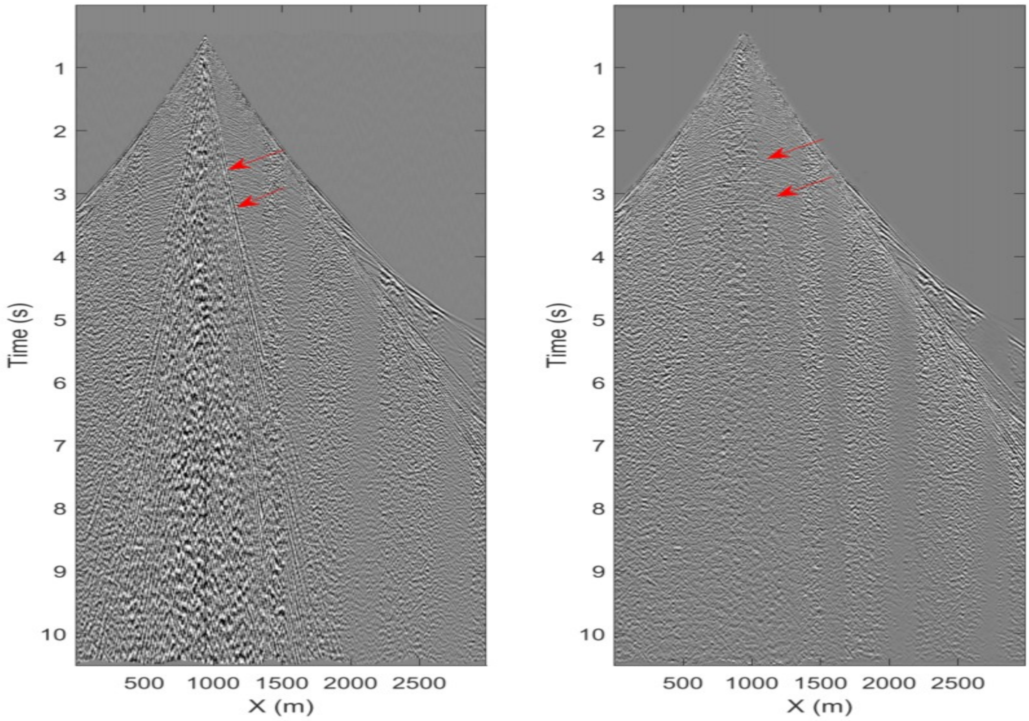


Figure 17 Scattered Ground Roll attenuation DL. The original noisy data set is shown on the left. The denoised dataset is shown on the right. The scattered ground roll is dislodged, as the red arrows indicate[3].

4.2.1 Deep Learning Methods

Traditional methods such as tomography and full waveform inversion (FWI) suffer from various bottlenecks, making seismic imaging a tough issue. This happens because of the curse of dimensionality that makes imaging time-consuming, as also to choose the suitable velocities, imaging cites keen on human interaction and at last because the data that were recorded lack low-frequency energy; nonlinear optimization demands good initialization or low-frequency information.

Seismic geophones are generally erratically spaced or not dense enough for Nyquist sampling because of environmental or economic restrictions. To improve inversion resolution, it is needed to rebuild or regularize the seismic measurements to a compact and standard grid[3].

To start with, recorded data are used as inputs and velocity models are used as outputs in end-to-end DL-based imaging approaches. This provides a distinct imaging methodology. DL methods dodge these limitations and provide a next-generation imaging technique. Initial trials of DL in staking[132], tomography[60] and FWI[133] present hopeful results on synthetic 2D data. A major issue lies in the fact that inputs are in data space and outputs are in model space, and this dual output result is accompanied by high dimension parameters. The U-Net is employed for deportation from various dimensional fields, while the down-sample is used to decrease the parameters while training the DNN. Figure 18 presents the results obtained by inverting the velocity[133].

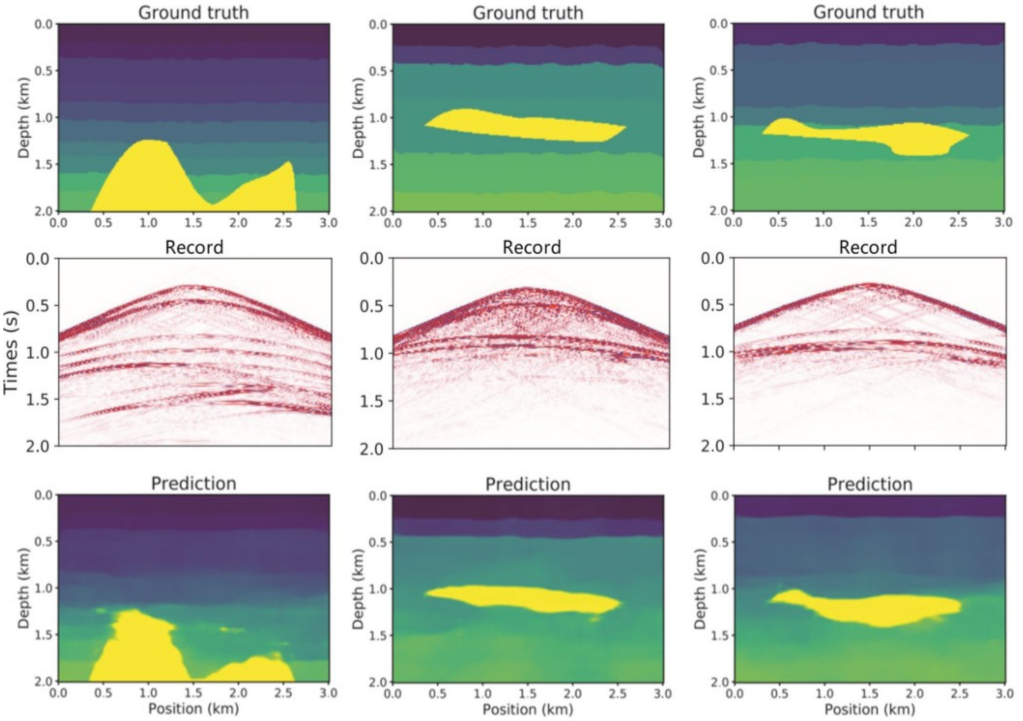


Figure 18 Using U-Net to forecast the velocity model from unprocessed seismic measurement (Yang & Ma, 2019). Several velocity profiles are shown in the columns [133]. The seismic records produced from a single shot, the anticipated velocity models, both the ground truth velocity models are displayed in order from top to bottom[3].

Though, there are also drawbacks to end-to-end DL imaging, like the absence of training samples and the limited sizes of the input data due to memory restrictions. One interesting paper generated a great number of models to build the training set by using smoothed natural images as velocity models. An example of transforming a color image with three-channels based on a velocity profile is shown in figure 19[134].

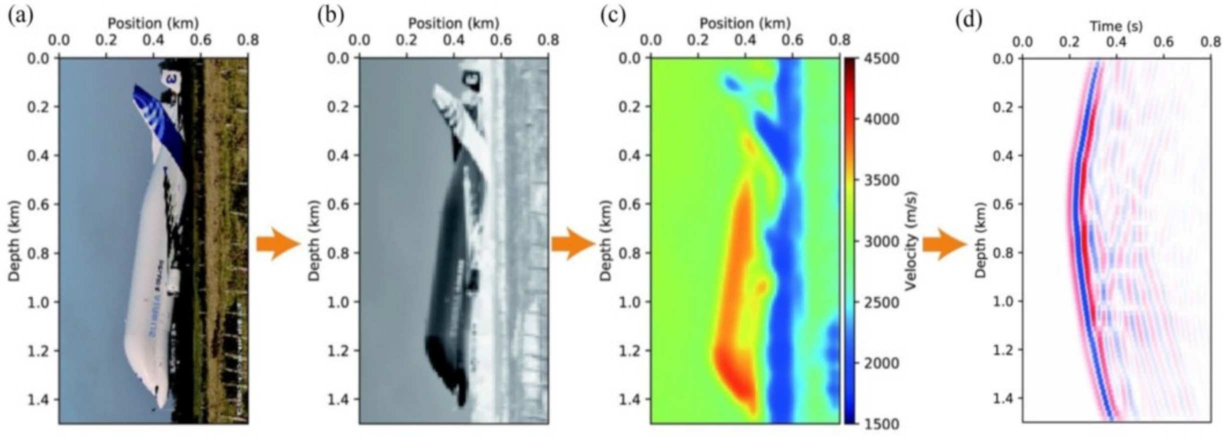


Figure 19 Conversion of a 3-channel color image to a velocity pattern[134]. (a-c) source image colored, corresponding velocity model, and the grayscale version of the image. (d) is an earthquake dataset made up of a cross-well structure on (c)[3].

In order to make DL-based imaging implementable to large data sets, more work is targeted in coworking with traditional methods and solving one of the bottlenecks mentioneed above, like the extrapolation of the frequency range of seismic data from high to low frequencies for FWI[135][136], and the addendum of restrictions to FWI[137].. In order to assuage the "curse of dimensionality" of the total amelioration in FWI, CAE is employed to decrease FWI's dimension based on ameliorating in lurking space.[138]. Another endeavor is concerned with forward modeling's high computing expenses when using the higher order finite discrepancy approach. GAN is deployed to generate high precision wavefields from low precision wavefields with low order finite discrepancy in the context of surface related propagators, ghosts and dispersion[139]. Figure 20 illustrates the fact that U-Net can be used for velocity sampling within stacks[140]. The input consists of seismological data while the output presents the constant value of one (1) when a seismic event is detected. All other cases of non-seismic events are represented by the constant zero (0).

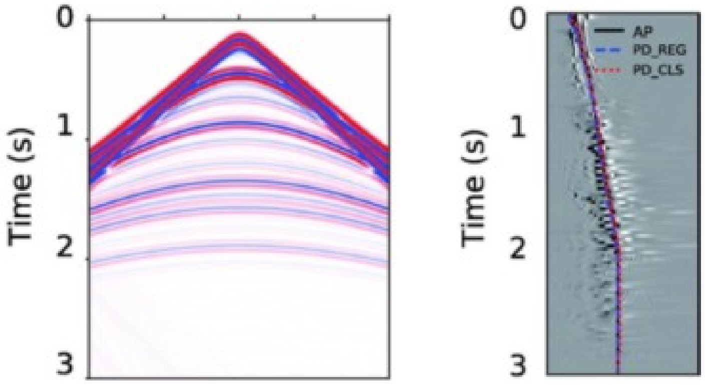


Figure 20 U-Net based velocity picking[140]. Seismological data are the inputs, on the left. The picking positions at the right represent the outputs. AP stands for approximate root mean square velocity. The expected velocity for the regression as well as the classification network is represented, accordingly, by PD_REG and PD_CLS.[3].

Alternatively the FWI algorithm is able to be replaced by an RNN loss function. The RNN architecture resembles a finite discrepancy in time progress alongside parameters of the network according to the chosen speed model. The optimization of an RNN is hence tantamount to the optimization of an FWI[141]. The concurrent inversion of velocity and density is an expansion of such a strategy. Figure 21 illustrates the structure of an amended RNN. It is based on the acoustic wave equation[142]. The chart is a flow chart of the discrimination of the wave equation in an RNN. Instead of using a gradient-descent-based approach, the ameliorated approach at FWI still has the possibility of being taught to a DNN and understood by it[143]. In an effort to adequately take into consideration the gradient's historical records based upon an RNN instead of hand-crafted directions, an ML-descent method is suggested.

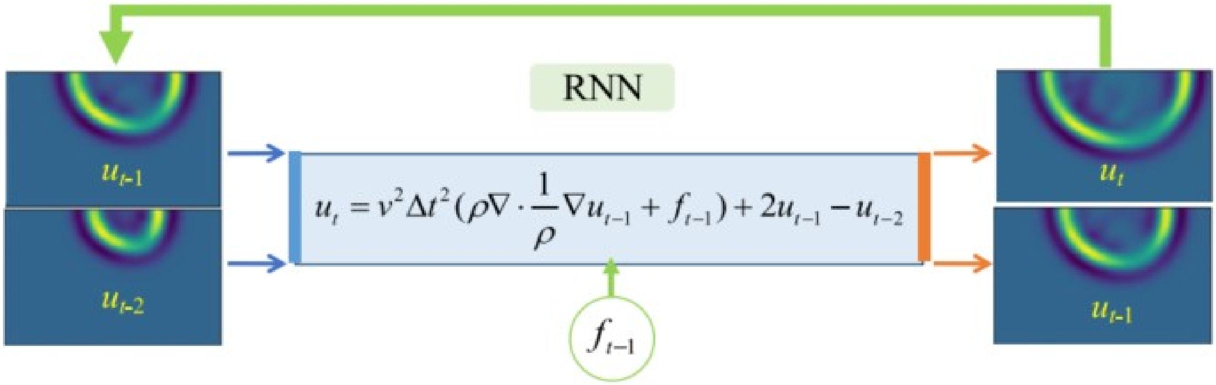

RNN
u_{t-1}
u_{t-2}
$u_t = v^2\Delta t^2(\rho\nabla\cdot\frac{1}{\rho}\nabla u_{t-1} + f_{t-1}) + 2u_{t-1} - u_{t-2}$
f_{t-1}
u_t
u_{t-1}

Figure 21 Changed RNN based on the equation of the acoustic wave to model waves[142]. The diagram is a representation of the discretization of the wave equation in an RNN. To effectively optimize velocity and density, the auto-differential mechanics of a DNN is used[3].

4.2.2 Seismic Interpretation

To excerpt subsurface geologic information and locate subsurface sweet spots, seismological interpretation (faults, strata, dips, etc.) or feature examination (impedance, frequency, facies, etc.) can be used. Though, both tasks demand expert intervention which makes them time-consuming. Initially, work indicates that DL appears to have the capacity to enhance the effectiveness and precision of seismological meaning and feature examination.

In the field of seismic interpretation, the process of identifying the position and characteristics of faults, layers, and dips can be likened to item tracing in computer vision. Accordingly, DNNs for imaging could be immediately implemented for seismological interpretation. Yet, obtaining a public training set or hand-crafted building a training set for domain data is hard, in contrast in the computer vision sector. It is more effective and can generate analogous results to produce realistic synthetic datasets instead of manual field datasets. In order to create a reasonably realistic 3D training data set, it is necessary to randomly select folding and faulting parameters within a reasonable range[144]. Subsequently, the dataset is utilized to train a 3D U-Net model, enabling the interpretation of seismic structural features such as faults, strata, and dips within datasets specific to the field. For small percentages of tracked objects, a class-balanced binary cross-entropy loss function is employed to ensure that the network does not solely predict zeros[145]. A different solution to an artificial training set is a semi-automated method where it annotates targets at an unrefined scale and projects them on a thine scale[146]. Figure 22 presents an instance of artificial post-stack image and region fault examination[144].

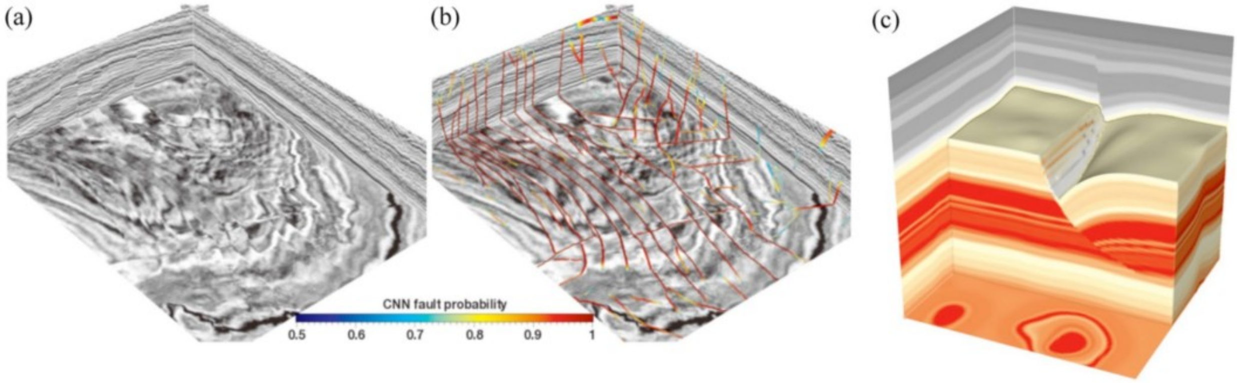


Figure 22 (a)A representation of post-stack dataset. (b) The outcome of (a)'s fault prediction. (c) present a synthetic dataset [144].

Attribute analysis is alike to image classification. The input is a seismic image, and the output is areas labeled with different features. Consequently, seismic attribute analysis can immediately apply DNNs for image classification[147][148][149]. A DNN can work in a cascaded manner if the features cannot be calculated immediately from the seismic data[150]. If labels are not provided, characteristics extraction is done using CAE, and unsupervised clustering is done using a method like K-means[62][151][152]. Unsupervised grouping of comparable features is mentioned to as clustering. For instance, it can be used for clustering to resolve if a region has river facies or faulting based on stacked segments. For optimal attribute extraction, CAE and K-means can be optimized concurrently[61]. A 1D CycleGAN-based impedance inversion method was suggested for the moderation of the reliance of vanilla CNNs on the number of specified earthquake data obtainable[153]. No pairing of training sets was demanded for the CycleGAN. Only two sets, one with high fidelity and one without, were demanded. An RNN is used in the facies analysis to take into account spatial continuity and resemblance between neighboring traces.[154].

4.2.3 First Arrival

First-arrival picking is utilized to choose the initial jumps of helpful signals. This process is fully automated. However, it requires strong hands-on involvement to validate selections, including important static corrections, low energy, low signal-to-noise ratios, and dramatic phase changes. Using realistic seismic data, DL assists in improving the automation and precision of first-arriving picks. One can naturally transform first arrival picking into a classification issue by regulating the first arrival to one and other places to zero when using DL[155]. Though, a regulation like this may have the effect of unbalancing the labels. An intriguing approach is to face the picking of the first arrival as an image classification issue, where everything in front of the first arrival is set to zero, and everything in the positions after the first arrival is set to one[146]. For noisy statements and scope data sets, this method works well. An extensively sophisticated picking method, as an RNN, could be used to benefit from the general information after the compartmentalization image is received[156].

4.3 Seismology through Deep Learning

Seisbench is an open source python toolbox for ML in seismology. This allows to analyze and modelize earthquake data automatically and take the outcomes we are looking for, without the need of us to arrange anything manually every time or giving wrong parameters[157]. The python libraries

Keras and TensorFlow provide many features to make this work easier for the user. Seisbench has already pretrained, mostly in PyTorch, six deep learning models that each of them has a certain task that will be more discussed forward.

An application of deep learning is presented in the subsequent example. Due to the independent samples in the classification data set, plain, densely joint feed-forward networks are employed. A CNN should ideally be fed image data or spatially linked datasets, whereas time series are frequently best handled by RNN. The TensorFlow library was used in the creation of this example.

All contemporary DL libraries use a modular approach to DNN construction, abstracting operations into layers. These layers can be bundled in extremely versatile and adaptable ways into input and output configurations. Below is realized the plainest architecture, is a sequential model, that is composed of one layer of input and one of output, along a "stack" of layers. In order to create very sophisticated neural network pipelines, it is feasible to establish increasingly complex models with numerous inputs and outputs as also layer branching. These kind of models are known as function API and sub-classing API, though they are not needed for this section. The model of the example is composed of Dense layers and a Dropout layer, that are sequentially ordered. As it appears in the following example, densely bonded layers comprise a determined amount of neurons with the proper activation function. Any neuron executes the calculation that it is described in the equation (1), where σ designate the activation. Sigmoid and tanh activations are infrequently applied in modern neural networks. Their activation characteristic led to a loss of information for great positive and negative values of the input, commonly referred to as saturation [158]. This saturation of neurons deters good attribution of DNNs until a new nonlinear activation function is substituted[159]. Because of their non-saturating qualities, the rectified linear unit (ReLU) is widely recognized as aiding the construction of very large DNNs.[160]. As it is shown in both figure 23 and equation 6, it zeroes all negative values and gives a linear response for positive values. Since it began, a large number of other rectifiers with different qualities have been imported.

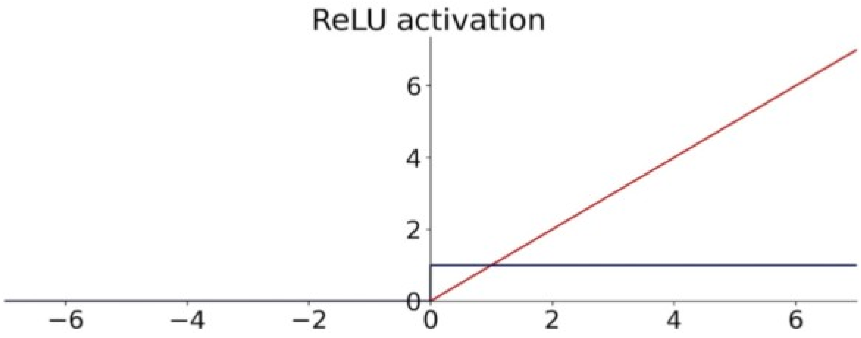


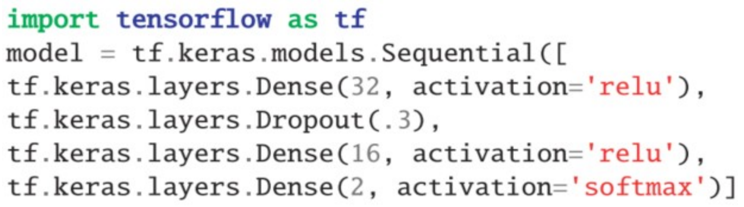
Figure 23 Effective gradient calculation using ReLU activation, which is represented by red and derivative which is represented by blue.[84]

$$\sigma(a)=\max(0,a) \qquad (6)$$

The "softmax" function on the output layer is the other activation function utilized in the example. Because it normalizes all the activations on all the outputs to a single one, this activation is widely employed for classification tasks. It accomplishes this by implementing the exponential function to each of the class C outputs and dividing the result by the sum of all exponentials:

$$\sigma(\vec{a}) = \frac{e^{a_i}}{\sum_{p}^{C} e^{a_p}} \qquad (7)$$

In addition, the example employs a Dropout layer, which is a common layer used for network regularization by putting randomly a particular proportion of nodes to zero for each repetition. Neural networks are specifically vulnerable to over-fitting, which can be mitigated by a variety of regularization approaches, including input data augmentation, noise injection, L1 and L2 limitations, or early-stopping of the training loop[161]. Yet, noisy student-teacher networks can be exploited for regularization by modern DL systems[162].



```
import tensorflow as tf
model = tf.keras.models.Sequential([
tf.keras.layers.Dense(32, activation='relu'),
tf.keras.layers.Dropout(.3),
tf.keras.layers.Dense(16, activation='relu'),
tf.keras.layers.Dense(2, activation='softmax')]
```

Figure 24 Dropout layer[84].

These sequential models, additionally are employing for simple image classification models using CNNs. Conversely of dense layers, these are constructed with convolutional layers, which are readily available in 1D, 2D and 3D as Conv1D, Conv2D and Conv3D accordingly. For the n m-dimensional image G, a two-dimensional CNN learns a so-called filter f, represented as:

$$G^{*}(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{m} f(i,j) \cdot G(x-i+c,\ y-j+c), \qquad (8)$$

which leads to the central outcome G* about the central coordinate c. Each layer in CNNs learns multiple of these filters f, which are frequently followed by a down-sampling procedure in n and m to condense the spatial information. This is used as a coercion function in succeeding convolutional layers to acquire more and more abstract representations as it is illustrated in figure 25.

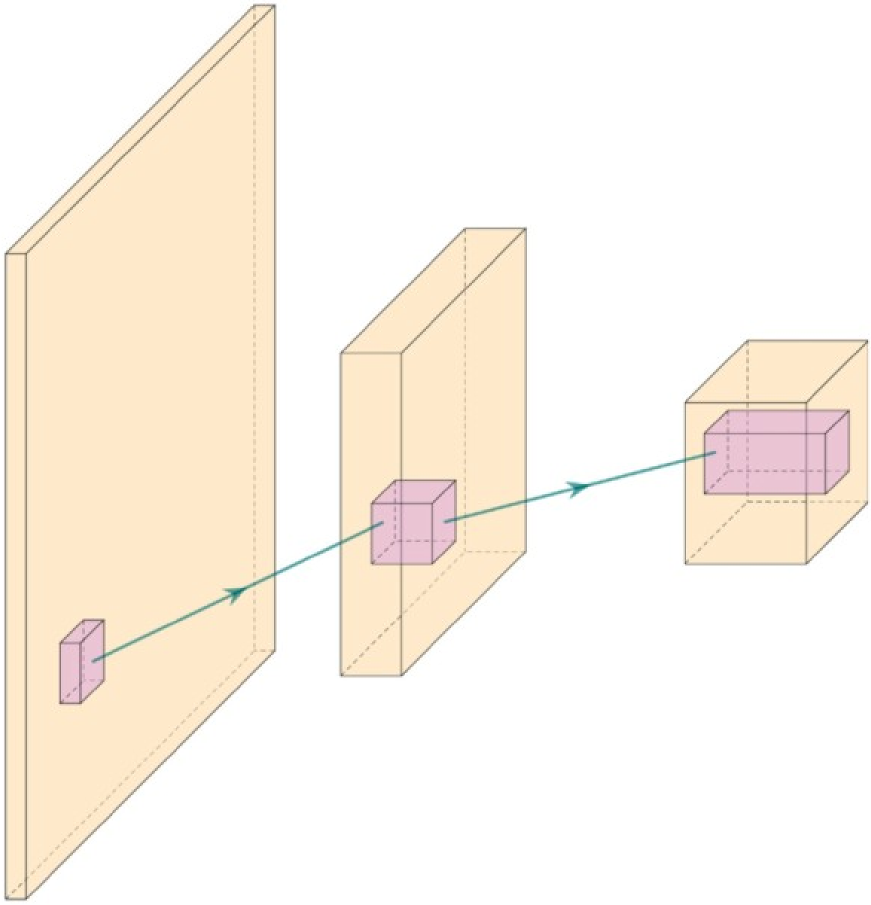


Figure 25 Convolutional network of three layers. Yellow presents the input image that is convolved with different firlters or kernel matrices, that are represented by purple. Usually, the convolution is employed to downsample an image in the spatial dimension, whilst the dimension of the filter response gets extensed at the same time, so the "thickness" in the schematic it gets extensed too. The ML optimization loop learns the filters. The communal weights inside a filter ameliorate the performance of the network compared to a classic dense networks[84].

This sequencial example model of densely joint layers with a solely input, 32, 16, and 2 neurons comprises altogether 754 trainable weights. Each of these weights are first assigned to a value that is pseudo-random and frequently selected from a distribution that facilitates quick training. In order to compare the outcome to the predicted values numerically, the data are sent across the network. This kind of training is stipulated as supervised training and error-correcting learning, where is a kind of Hebbian learning. Different forms of learning that exist and are used in ML, is for instance competitive learning in self-organizing maps.

$$\mathrm{MAE} = |y_{j\,-}\, o_j| \qquad (9)$$

$$\mathrm{MSE} = (y_{j\,-}\, o_j)^2 \qquad (10)$$

Usually, in regression issues, the errors are calculated by using the mean absolute error (MAE) or the mean squared error (MSE), of L1 is presented in the equation (9) and L2 is norm is presents in the equation (10), accordingly. Classification issues are a specific type of problem that can exploit a different type of loss called cross-entropy (CE). The cross-entropy depends on the real label y and the prediction in the output layer.

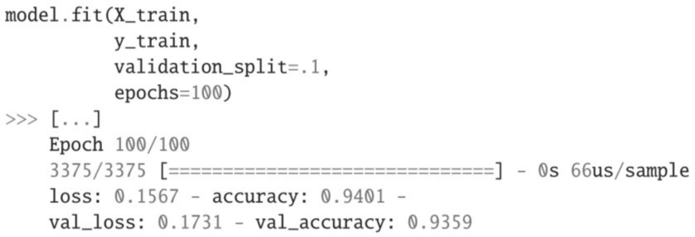
$$CE = -\sum_{j}^{C} y_j \log(o_j) \qquad (11)$$

Several ML data set have a single true label $y_{true}=1$ for a class $C_{j=true}$ letting the rest $y_j=0$. This renders the total of all labels obsolete. Although it is arguable how much binary labels resemble reality, it simplifies equation (11) to minimizing the (negative) logarithm of neural network output, as well known as negative log-likelihood:

$$CE = -\log(o_j) \qquad (12)$$

In technical terms, the data is a binary classification issue, which allows us to use the sigmoid activation function in the final layer to optimize a binary CE. This can expedite the computation, however this example is a demonstration of an approach that can be used for several different problems and is therefore applicable to the reader data.

The training of big neural networks can be excessively pricey with important developments in 2019/2020 reference to language models with billions of parameters (Google, OpenAI) trained for weeks on massive hardware infrastructure with a solely epoch taking many hours. This requires a validation of the unsightable data at the end of each epoch of the training run. As a result, neural networks, like all ML models, are typically trained with two sets of holdouts, a validation set and a final test set. The validation set can be given or it can be specified as a propotion of the training data, as it is following presented. In the instance, the training data set is decreased from 3750 to 3375 sole samples by holding 10% of the training data for validation later each epoch.



```
model.fit(X_train,
          y_train,
          validation_split=.1,
          epochs=100)
>>> [...]
    Epoch 100/100
    3375/3375 [==============================] - 0s 66us/sample
    loss: 0.1567 - accuracy: 0.9401 -
    val_loss: 0.1731 - val_accuracy: 0.9359
```

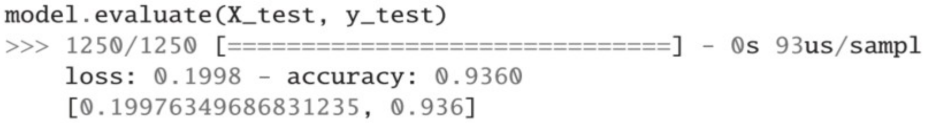
Figure 26 Neural Network[84].

Stochastic gradient descent (SGD), an incremental edition of the conventional steepest descent algorithm, is used to train neural networks. It is employed the Adam optimizer, a fast-convergent SGD variation, but a thorough discussion is not needed as it would be long and not necessary. In essence, Adam's optimization keeps a per-parameter learning pace of its first statistical moment (average). This is advantageous for sparse problems. The second moment (uncentered variance) is advantageous for noisy and nonstationary problems[163].

SGD with Nesterov impetus[164] is an optimization technique that simulates conjugate slope methods (CG) without requiring the excessive and complicated computations that are required for the search in CG. It is also the primary alternative to Adam. The problem with SGD is that converges in a greatly slower rate than Adam, while unofficially it finds a better optimum spot for neural networks.

Except of the loss value, it is presented also the accuracy metric. Accuracy provides a reasonable primary assessment of the number of samples that are correctly predicted with a rate amid zero and one, although it should not be the sole determinant of model attribution. DL models, unlike scikit-learn, are compiled after they are defined in order to be optimized on the available hardware. Thereafter, the neural network can be adapted, using the X_train and y_train data, just as the SVM and random forest models previously.

Moreover, a number of epochs can be specified for execution, as also other parameters that are default for the example. The number of epochs determines how many optimization cycles are conducted on the entire training data set. Traditional training of neural networks should always have more epochs than ML researchers originally appraisal. It can be tricky to determine in neural networks, all the sources of randomness and stochasticity, for research and examples to be reproducible. This example does not determine these so-called random seeds, because it would detract from the example. That entails that the outcome for the loss and accuracy will vary from the printed examples. Seed fixation is very significant in research to ensure reproducibility. Furthermore, statistical resolution of manifold fixed seeds is a great practice for referring outcomes in any ML model to avert bad practices or the so-called lucky seeds.



```
model.evaluate(X_test, y_test)
>>> 1250/1250 [==============================] - 0s 93us/sampl
    loss: 0.1998 - accuracy: 0.9360
    [0.19976349686831235, 0.936]
```

Figure 27 An example of model evaluation[84].

In the previous example, the SVM and random forest classifier were evaluated on unsight data. This is equitably significant for neural networks. Overfitting is a problem with neural networks, which it is aimed to be avoided by regularizing the weights and assessing the finished network on an unsight test set. The prediction on the test set is highly close to the latest epoch in the training loop, which is a great sign that this neural network generalizes to previously unsight data. Also, in figure 28 the loss curves don't converge very quickly as they converge. Though, it appears that if the training was continuing the network would overfit. In figure 29, the model decision boundary models too closely the local distribution of the data, which applies to the whole decision volume [165].

These examples are presenting the open-source revolution in ML software. Implementing several ML algorithms to scientific data looks trite with the consolidated API and utilities. This is evidenced by the recent explosion of applied geoscience ML publications. The ability to simply install a package and call model.fit(X, y) has replaced the necessity to implement algorithms. To assure valid, reproducible, and scientific results, these evolutions demand rigorous validation of models. Without this attentive validation, these contemporary tools can outcome in serious misinterpretation of results and, in fact, erroneous conclusions.

On the whole, today's neural networks profit from the evolution of nonsaturating, nonlinear activation functions. To speed up the training of DNNs, developments in stochastic slope descent with Nesterov impetus and the Adam optimizer (after AdaGrad and RMSProp) were indispensable. Training times are further decreased by taking advantage of the graphics hardware accessible on most high end desktop computers, which is niche for linear algebra computations. Finally, shallow and deep ML have become accessible to those who are not experts through open source software that is well-preserved, tested, and factualed with a consistent API.

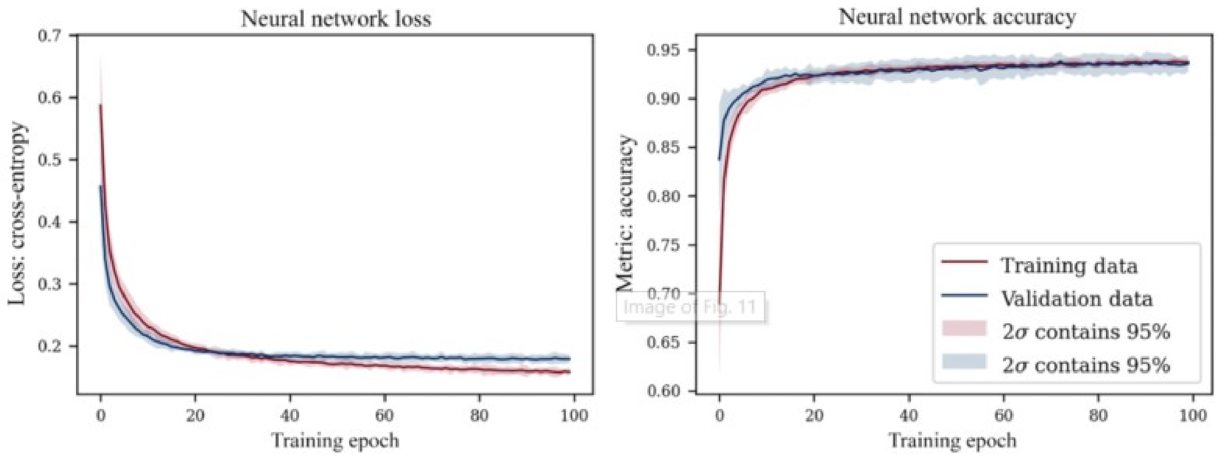


Figure 28 Instance neural network accuracy and loss upon 10 random initializations. Training for 100 epochs while displaying the confidence intervals of 95% both the loss and metric within the shaded area. It's crucial to analyze loss curves in order to assess overfitting. Overfitting is indicated by the training loss falling while the validation loss is almost at a plateau. In general, it is clear that the model has converged and is only showing slight improvements while running the risk of overfitting.[84].

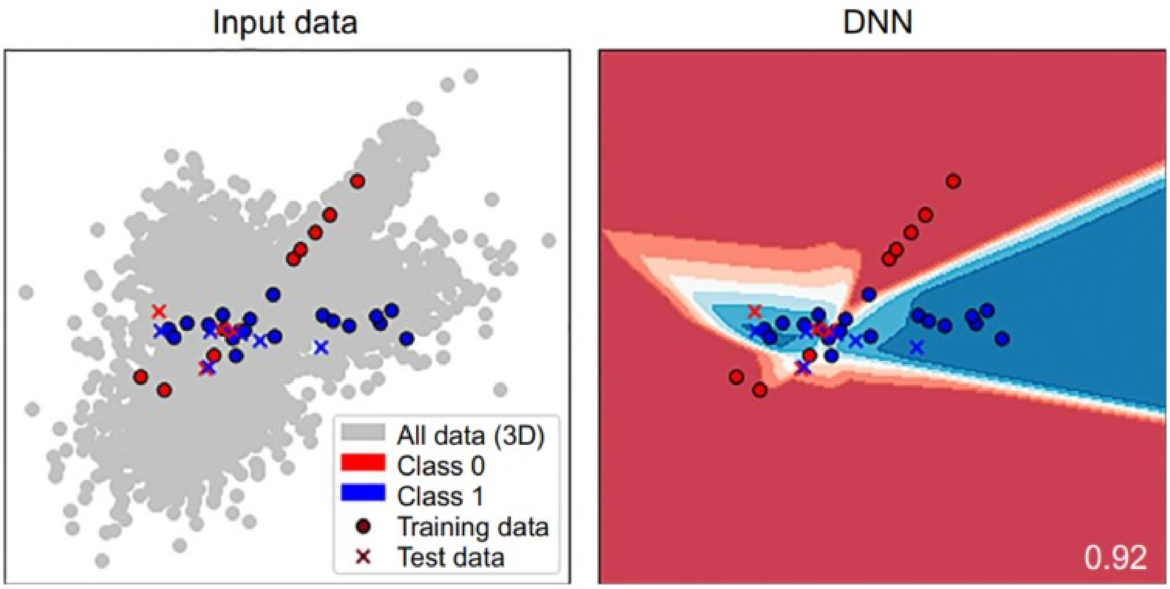


Figure 29 Central 2D intersection of the decision limit of a DNN trained on data containing three informative characteristics. The 3D volume is accessible in Dramsch [165].[84]

4.3.1 Earthquake Deep Learning Models

TensorFlow is an open source AI framework and Python library by Google, its main functions are complex and multi layers computations. It is designed for research and product build out, its operation adopts the data flow graph method, a method that is demonstrated by mathematical reckonings. TensorFlow supports Keras library for machine-deep learning purposes. Nowadays is mostly used the tf.keras API, interface (tf for TensorFlow), an integrated version of the initial one. It provides a flexible architecture that it allows to run and carry out calculations on one or more CPUs or GPUs with only a single API[50][166].

Keras is a Python deep learning library, which acts on TensorFlow framework as an interface[49][50]. It depends on NumPy, Pandas, Scikit-learn, Matplotlib, SciPy and Seaborn libraries[167].

BasicPhaseAE, DPP, GPD and PhaseNet models are ideally for phase picking, CRED works well for Earthquake Detection, while EQTransformer is perfect for Earthquake detection and phase picking. So depending on the application that is needed to be completed every time, there are different pretrained models, so it is not necessary to build a new model from the scratch, if this is not the case.

EQTrnsformation, it already exists in seisbench in TensorFlow and Keras libraries except of Pytorch as are the other models.

Some of the main issues with which ML helps seismology is in earthquake phase identification, in earthquake detection, as also in magnitude estimation and in EEW[157][168].

4.3.1.1 SeisBench's Models

The models that SeisBench's primary implementation used are mentioned beneath. For a more thorough explanation consult the reference section[169]:

- BasicPhaseAE[170], essential CNN UNet, was first applied to a regional aftershock sequence in Chile.
- CRED[171], a CNN-RNN Earthquake Detector, was trained on 500,000 training signals and noise examples from Northern California.
- DPP[172], DeepPhasePick, is a combination of a CNN for the detecting phase and two RNNs for determining onset time. The networks are designed for local event detection and picking, similarly to BasicPhaseAE and its first application on a regional seismic network in Chile.
- EQT[171], EarthQuake Transformer, an Attention-based Transformer Network used for picking and detecting events.
- GPD, Generalized Phase Detection[173], is a seismic phase detection CNN algorithm.
- PhaseNet[174], CNN autoencoder algorithm, customizes the U-Net compartmentalization framework to solve the 1D seismic phase classification issue.

4.3.2 Datasets

Seisbench provides datasets, for its users to be able to work with. These datasets are ETHZ, GEOGON, INSTANCE, Iquique, LENDB, SCEDC, STEAD and NEIC.

- ETHZ benchmark dataset was drafted by hand. It includes seismicity that has been locally to regionally documented in Switzerland and adjacent frontier regions. The Swiss Seismological Service (SED) at ETH Zurich registers the data on networks that are open to the public. Through SED's FDSN web service (http://www.seismo.ethz.ch/de/research-and-teaching/products-software/fdsn-web-services/), it's been acquired the waveform recordings as well as the related metadata information to build this dataset. The phases of any seismic event recorded by this network have been manually labeled, including the differentiation between the early and later phases (e.g. Pn vs. Pg).Together with the standard phase identification, information on magnitude and polarity is also given. For this dataset, a total of 57 metadata variables are available. We choose to integrate all M >1.5 events that occurred between 2013 and 2020. There are 2,231 events with 36,743 waveform examples in total. Every trace is a raw count. This dataset is getting separated the training examples. 61.1% covers the training examples for all the events before 1st of August 2019, 9.9% covers the development split for all the events from 1st of August to

4th of September 2019 and the rest 28.5% covers the testing split for all the events after that date.

- GEOFON, contains teleseismic dataset, that involves 2270 events which are including ~275,000 waveform examples for 2009-2013. Waveforms from more than 800 seismic stations around the world are obtained and parsed by the GEOFON monitoring Service. The magnitudes are approximately between M 2-9. The majority of events compromise mid-sized to large events M 5-7, as shown in figure 30. Possible peripheral events with smaller magnitudes originate in parts of Europe and northern Chile[168]. It is important to be noted the different class distributions of the chosen phase types for the GEOFON dataset. S onsets have been chosen for local and nearby regional events, and for a tiny portion of those, Pn and Pg are also included. There haven't been many S onsets chosen for teleseismic occurrences. There have been sporadic but incomplete selections of depth phases. The dataset it's been split into 3 parts: 58.6% in training examples, for all the events that have been occurred before the 1st of November 2012, 10.1% in validation examples that are refered to all the events that have happened between 1st November 2012 and 15th of March 2013 and finally testing examples are covering the 31.3% for all the events that have happened after that date.
- INSTANCE benchmark dataset[175] contains ~130,000 samples of noise in addition to ~1.3 million regional 3-component waveforms from ~50,000 earthquakes between M 0 and 6.5. It is offered distinct access to each of the partitions of this dataset in SeisBench. The seismic events are further separated into datasets with waveforms in counts and with waveforms in ground motion units. Both noise examples and signal examples are provided as separate datasets. There is also a composite dataset that includes all waveform and noise instances in counts. There are 115 different metadata variables offered. This dataset offers a wealth of derived metadata in addition to the conventional metadata variables, like peak ground acceleration and velocity. In this case the dataset is separated randomly selected "event-wise" in 3 parts. So all the waveform examples that are owned to the same event, are in the same group. Therefore, 60.3% covers the training, 10% covers the validation and 29.7% of the dataset, cover the testing.
- The Iquique benchmark dataset was first used to train the DL picker[170]. It consists of locally registered seismic arrivals from all over northern Chile. The dataset is associated with 23 metadata variables, and all waveform units are represented as raw counts. The training examples are randomly sampled to determine the training, validation, and testing splits for this dataset, yielding 60%, 30%, and 10% for each split for training, validation, and testing, accordingly[168].

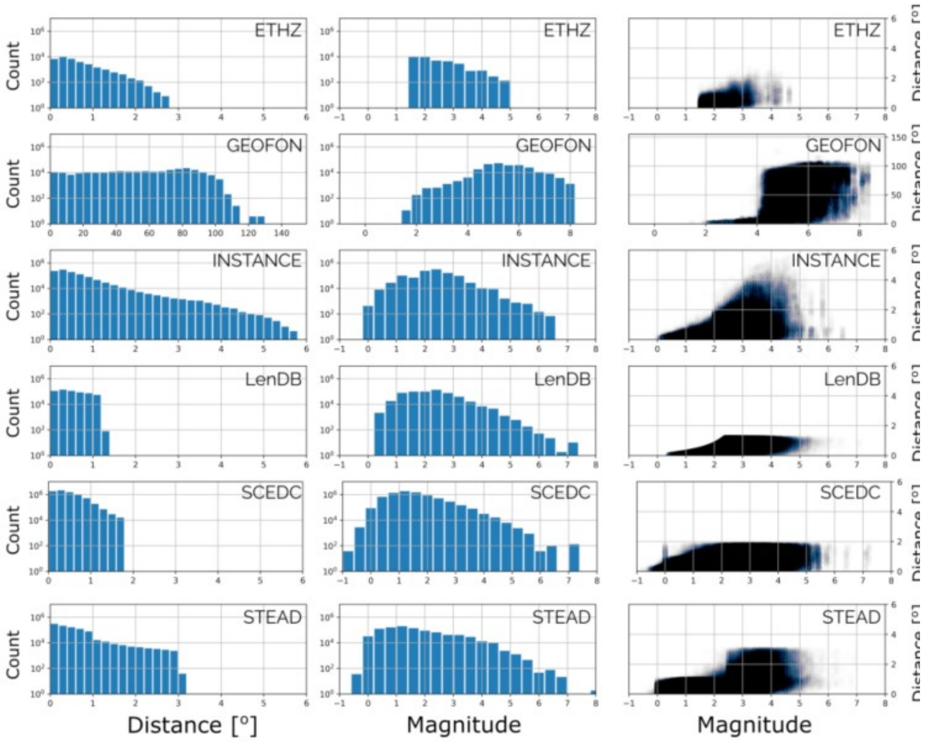


Figure 30 Epicentral distance and magnitude distributions via logarithmic histogram form for datasets that include source and station data. All points are plotted with transparency in the 2D scatterplot according the last column to emphasize the overall distribution. The magnitude or source both station location data for the Iquique, NEIC, GPD, JGRPick, JGRFM, and Meier2019JGR datasets are missing. Therefore, these datasets are not displayed.[168].

- LENDB benchmark dataset[176] is a published dataset, that is consisted by local earthquakes records in a worldwide set of 1487 broad-band and very broad-band seismic stations. The dataset includes ~1.25 million waveforms, and it is separated into local earthquake and noise examples, of 629,095 and 615,847, respectively. In order to translate the recordings into physical units of velocity, the data were processed using a bandpass filter between 0.1 and 5 Hz and the instrument response was deconvolved. LENDB exclusively offers automatic P-phase picks, in contrast to the other datasets. There are a total of 23 metadata variables in this dataset. All cases with waveform start timings prior to 16th of January 2017, are chosen as training examples, accounting for 60% of the total dataset. Examples falling between this date and the 16th of August 2017, which comprise the validation split at 9.5%, and any examples falling after that date, which comprise the test split at 30.5%, respectively.
- The Southern Californian Earthquake Data Center (SCEDC)[177] was built from publicly accessible seismic waveform data. The Seismic Transfer Program (STP) client[178] is used to receive the waveforms and related metadata. All events were manually selected from the received seismic arrivals. Are chosen all publicly accessible recordings of earthquake events in the Southern California earthquake grid for the period 2000 to 2020. Solely local recordings of seismic events are considered approximately between M -1 to M7, with source to station aisles spanning for a maximum distance of ~200km. The dataset is consisted by ~8 million waveform examples, that comprise ~7.5 million P-phases and ~4.3 million S-phases. This dataset also compromises a collection of seismic instrument

types, including: very short period, short period, very broadband, broadband, intermediate band, and long period instruments are also included, as are single and three-component channels. The examples' units are raw counts. The separation for this dataset is defined randomly, by 60% for the training, 10% for the development and 30% for testing.

- The STanford Earthquake Dataset (STEAD)[123], is a published benchmark dataset that includes a number of local seismic signals, seismic as well as nonseismic, together with noise examples. This dataset contains ~1.2 million waveforms, of which ~200,000 are noise examples and the rest of them involve seismic arrival from ~450,000 earthquake between ~M -0.5 and M 8. There are 40 metadata variables related with this event, and the units for the waveform examples are raw counts. The split of this dataset, is the same as in[10], where the separation happens randomly, with a training set consisted by 10% of it, afterwards it is added a validation set with random sampling from the rest of the samples. So, the final analogies are 60%, 30% and 10% for training, validation and testing, accordingly. These are some examples of situations in which publicly available waveform data and metadata were available for ML model training but some of the customary metadata was missing.
- The National Earthquake Information Center (NEIC)[179] published benchmark dataset includes ~1.3 million both seismic phase arrivals and universal source-stains aisles. Because the record lacks trace start time and station information, it is saved as a SeisBench format, but without the normally demanded information. The initial publication provided randomly sampled divides based on event-id, for the separation of training, development and testing set. This approach of random separation is applied in the SeisBench conversion of this dataset, same as before 60% for training set, 10% for development and 30% for testing.

SeisBench includes additional datasets that have been used to train noteworthy seismic DL algorithms. Because the waveforms for these datasets were usually pre-processed for training, including windowing and labeling, the initial station metadata for each training example is not available. Because several of the datasets employ selected waveforms from the SCEDC network, there is a possibility of metadata parameter and waveform overlap among the datasets listed below. The only variations are potentially different metadata variables among datasets, as for instance picked phase labels versus first motion labels.

The converted to SeisBench format, DL training dataset, include: the 'GPD' training dataset[91] including 4,773,750 examples of 4s waveforms, sampled at 100 Hz; the 'JGRFM' dataset was utilized for training the DL on basis of first motion polarity tracing routine in the study of Ross[90], including 6s Z-component waveform samples from 100Hz instruments; the 'JGRPick' dataset was utilized for training the DL-based picker; The 'Meier2019JGR' dataset, that included the S. Californian constituent of the training examples from the work of Meier[180].

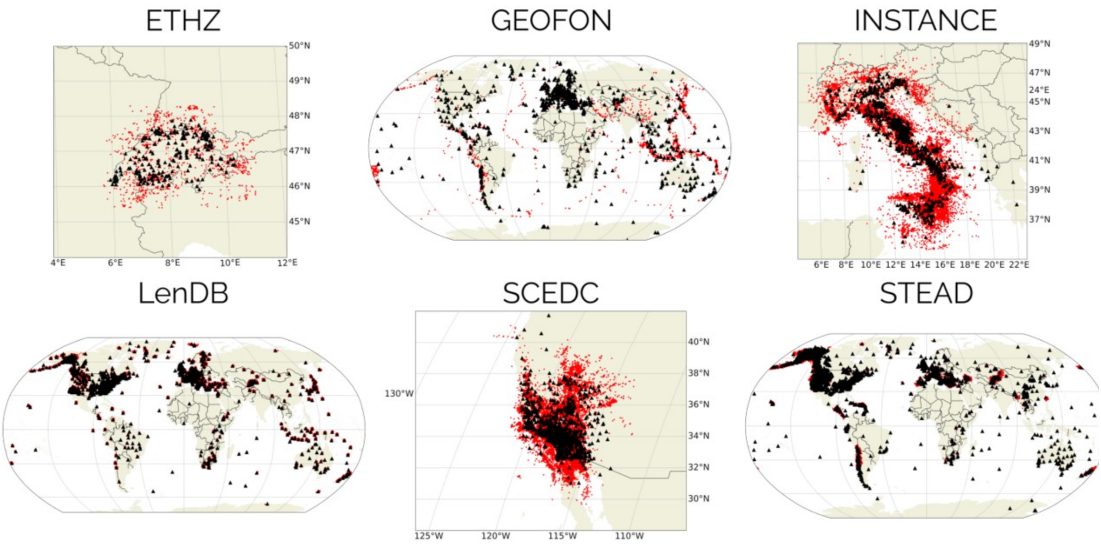


Figure 31 Benchmark datasets were incorporated into SeisBench at the time of the software's original release; seismic sources are represented by circles, and stations by triangle markers. Some further datasets contained in the SeisBench initial release dataset collection are not displayed because they either lack source information (NEIC, GPD, JGRPick, JGRFM, Meier2019JGR), either have insufficient event data for charting, or both (the local Iquique dataset)[168].

5 Earthquakes from Deep Learning's perspective

Dl algorithms that are based on DNN layers, come to take advantage of seismic data structures. These algorithms learn a map to the target distribution of interest and extract beneficial elements and visualizations from seismic waveforms, including the probability of distinguishing seismic from nonseismic signals. Large seismic datasets such as STEAD[123] have been developed for research based on DL since the performance of a DNN enhances with the abundance of different training samples.

5.1 Data process and training

Not every problem has a large-scale dataset, for instance the datasets that pertain to very large earthquakes are limited, which are -fortunately- rare due to the power-law distribution of earthquake magnitudes. On the other hand, the construction of a high-quality training dataset, along with adequate labeling and quality control, needs a great amount of work and time. Data augmentation, which includes several methods to generate additional training samples based on gathered datasets to broaden the amount and variety of training samples, is one way to get around these issues. The abeyant for applications on seismic datasets is demonstrated by the success of data augmentation in preventing overfitting and enhancing the generalization of DL models trained on tiny training datasets.

The term "generalization" is frequently used to describe the process of realizing that a particular attribute falls within a larger category. A trained neural network's capacity to perform well on data that were not included in its training or validation is referred to as "generalization" in DL. Examples of such data include a singular seismic source or seismograms from an area that was not used in training. The network architectures, optimization techniques, and training datasets are some of the elements that influence generalization. The training datasets' size, label correctness, and completeness are crucial factors in creating an effective model. Any or all of these characteristics might be absent from a training dataset; in this case, data augmentation might be a good way to boost a model's performance.

In the high-dimensional data space, as shown in figure 32, only a small portion may be covered and may contain weak limits on the neural network's potential decision boundaries learned by the neural network from the contents of training samples, including signals and noise. This results in meager achievements, in the form of low true positives or high false positives. The goal of data augmentation is to raise the complexity and size of the training sample, which broadens the sampled feature space and enables the neural network to learn a better decision boundary that will decrease false positives and false negatives and enhance generalization on unknown samples.

DNNs are frequently trained with data augmentation to improve performance in classification and recognition challenges for computer vision[55,181,182,183], audio processing[184,185] and other areas[186,187,188]. With proper implementation of data augmentation, the risk of overfitting is reduced. This is achieved by increasing the number and variability of training data. The majority of data enlargement methods are created for images since they can readily maintain semantic meaning. For applications with little labeled data, it can be particularly effective.

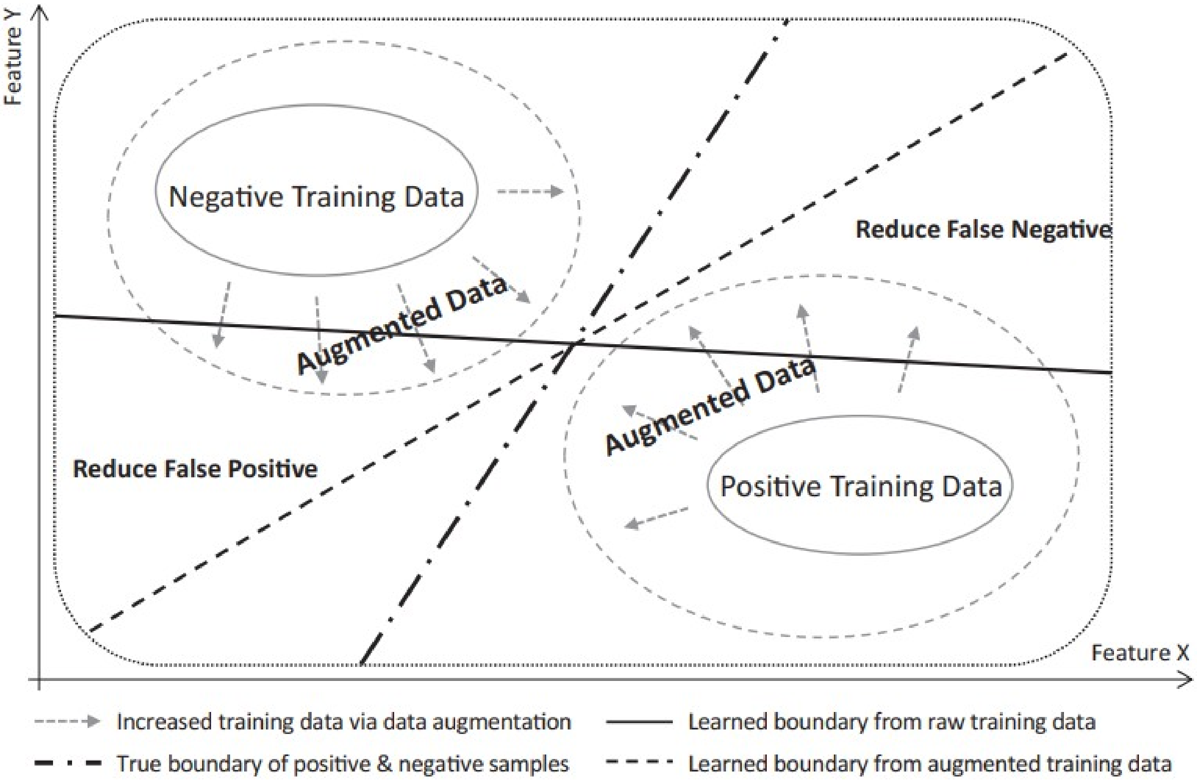

Feature Y
Feature X
Negative Training Data
Augmented Data
Reduce False Negative
Augmented Data
Reduce False Positive
Positive Training Data
Increased training data via data augmentation
Learned boundary from raw training data
True boundary of positive & negative samples
Learned boundary from augmented training data

Figure 32 The dataset enhancement increases the data space (small dashed arrow) spanned by the collected seismic signals and noise, and improves the bounding of the decision limit (change from solid to dashed line).By expanding the data space through data enlargement, trained DNNs can better generalize to unseen signals and noise, with reduced false negatives and false positives.[189]

Yet, some enhancements that are suitable for vision, such as horizontal flipping, rotating, and shearing, are not suitable for seismic data because they breach the physical characteristics of the relevant waveform data. Studies on seismic data augmentation methods are quite rare. In order to avoid the neural networks from picking up and memorizing any artifacts in how the data used for training are arranged, clear augmentations are required because seismic data are typically collected and organized in a conventional manner, thus using a defined time frame around the P-wave arrival. Random shifting, regrouping events and noise, and channel (station) dereliction, are less obvious enlargements. They are able to abated bias in training data and to raise model attribution, as they consistent with the feature of seismic signals. The ability to apply the generalization of the model to scenarios of interest, such as low quality data, complex noise, and closely recorded events in earthquake swarms, is improved using the strategies that are demonstrate and discuss in this study. The examples show that, with the right augmentation, DL can be applied to smaller datasets for earthquake monitoring than would normally be feasible[189].

5.1.1 Data process with augmentation

To demonstrate the use of augmentation for training DL models on seismic data, it's been gathered a small, high-SNR training dataset with precise hands-on labels of 500 earthquake waveforms from the Northern California Earthquake Data Center (NCEDC)[190] registered before 2018. To identify the best model for training, it had similarly built a validation dataset with 500 additional high SNR earthquake waveforms from before 2018. With a considerably bigger test dataset consisted of 10,000 earthquake waveforms captured in 2018, it was assessed the augmentation techniques. This ratio selection for the datasets of training, validation, and test was made with the specific intent of assessing the impact of augmentation on small datasets. The training dataset would require a bigger sample size than the validation and test datasets in order to be practical for real applications. In figure 33 is illustrated the allotment of SNR of the data. Below the standard deviation of waveforms preceding and following the earliest manually selected P-wave arrival are used to determine the SNR. Figure 33D displays the allotment of epicentral distances for the test dataset, with the majority of source-station lines falling within the 0 to 120 km range. The distributions of epicentral distances in the training and validation sets, they are not displayed, are comparable. It's been examined the implications of data enlargement for deep neural network training using the now extensively researched phase choosing problem as an example. It was employed the same NNA as PhaseNet[174], which is a fully CNN built for seismic phase picking. We eliminated abandonment, rate of learning decay, and weight decay (regularization) from the design and kept only batch normalization to prevent the complicated impacts of hyperparameter customization.

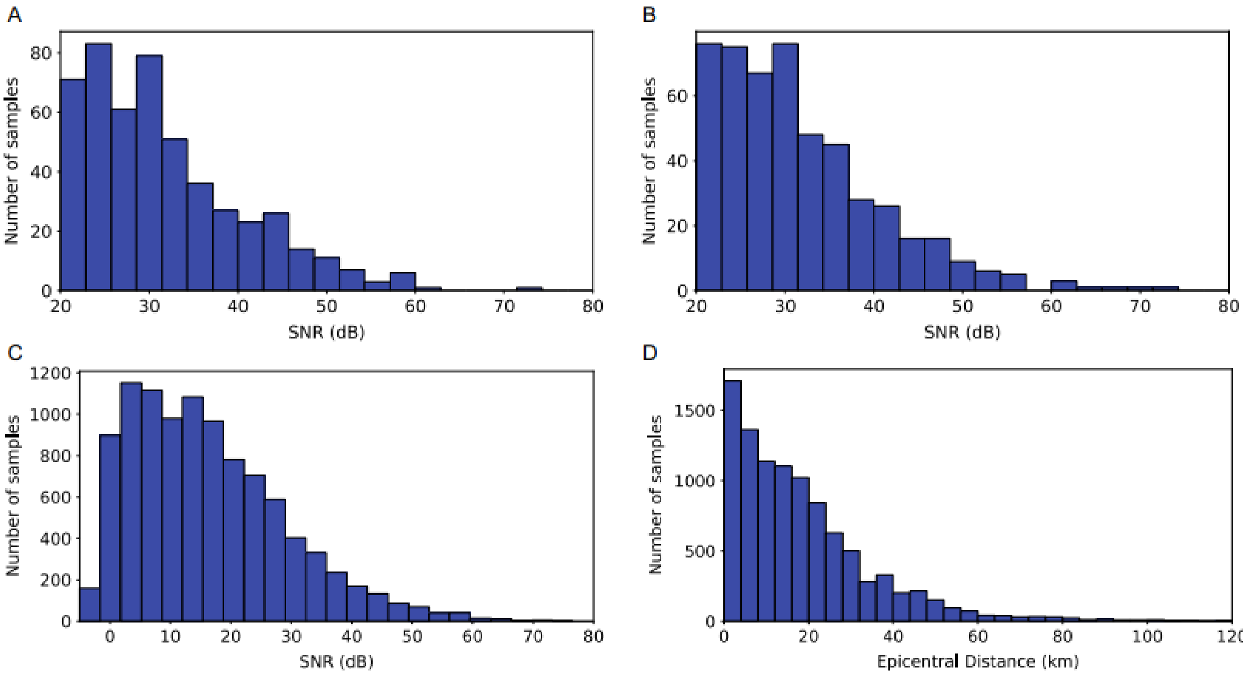


Figure 33 Statistical analysis from the benchmark datasets include the SNR allotments of (A) 500 samples of high-quality for training (>20dB) and (B) 500 samples of high-quality for validation (>20dB), both taken from earthquakes that happened before 2018; (C) the SNR allotment of 10,000 test samples taken from earthquakes that happened in 2018; and (D) the epicentral distance allotment of the test samples. Epicentral distance allotments from the training and validation datasets are comparable; they are not displayed here.[189]

The Adam optimization was used, while the learning rate was set to 0.01, the batch size to 20, and the total number of epochs to 100. By creating a new, large number, training sample and using enlargement, the training set size is increased. In this example, to achieve an increment in the diversity of training samples in each epoch with enhancement, the total number of training samples was kept the same in each epoch, at 500 samples, and on the fly data enhancement was applied. The ability to generalize for various issues and data formats will also depend on the neural network architecture (NNA), the training process, and the optimization technique[191]. In this chapter, the training procedure is kept plain and the focus is on the impact of data growth, for seismic data. The findings should be applicable to related issues including earthquake detection, phase detection, and other DL applications to seismic data with appropriate adjustment.

5.2 Increments

The fields of image and acoustic signal processing have seen the application of numerous data enlargement techniques. In this section, are studied the enhancements that are particularly relevant to seismic data processing. It is important to be noted that the reported performance will have some randomness due to the randomness in neural network startup, optimization, and data replenishment.

5.2.1 Arbitrary alteration

The common method for gathering seismic training data is to use the phase arrival data from earthquake catalogs. Thus, it is enticing to establish a cutout window based on the time related to a specific seismic phase, like the arrival of the first P-wave. Positional bias can be introduced into the model when neural networks are trained with data with a fixed allusion time or with a small time shift of a few seconds around the reference time point. Therefore, instead of learning more broad functions from the feature space, a neural network model of this type is more likely to learn by heart the location of the anchor time point. Hither, by using no random alter, a limited random alter from 10 to 15s, and a full random alter from 0 to 30s, it gets trained three models on data with a 30-s time window. Each training waveform is randomly shifted in real time, resulting in a unique shift for each epoch. By moving the test waveform from the left to the right side of the window, we can analyze the P-wave arrival predictions and document the estimated activation scores at the actual P-arrival sites, as it is illustrated in figure 34. As it’s shown in figure 34A-C, the neural network persists to predict a high likelihood rating at 10s, which is the fixed P-wave arrival time throughout training, for the model trained without random shift, irrespective of where the genuine P-wave arrival. A presumption that phases are predicted to appear inside a specific window may be seen in the results for the model trained with a limited shift of 10-15s. Figure 34D, at the time window's edges, the anticipated activations decrease. If the test dataset is changed from 10 to 15 seconds inside the same assessment window, this bias can be disregarded; however, when used to continuous data where the signal time of arrival is unknown beforehand, the performance will suffer. A model may be useless for identifying earthquakes beyond the shift window employed in training if it has a very small shift range. The instance with the random alter from 0 to 30s, in comparison, shows strong activation scores over the whole frame. Based on a certain window size, this possible bias from random shift may also appear in other DL applications, such as earthquake detection. Random alter can increase sample variation, enhance detection performance, and mitigate potential bias across the prediction window. A predetermined set of shifting periods are applied to the 500 training examples in Table 3, while random alter is applied on the fly such that every sample is shifted differently at every epoch. Better performance is achieved by allowing for more time shifting diversity across various training epochs through random alter on the fly. A fixed set of time altering for all epochs, on the other hand, may only sample a restricted number of time points, especially for a small quantity of training data. To circumvent the need for zero padding and efficiently implement random alter on the fly throughout the 30s window in this scenario, we trimmed the training sample to a bigger window of 90s. By having the neural network learn the transition from zeros to signals and using this artifact as the basis for its predictions, zero padding may induce a minor bias. For DL-based seismic phase detecting/picking models, investigating the allocation of arrival times in a training dataset ahead of and following enlargement is a recommended technique.

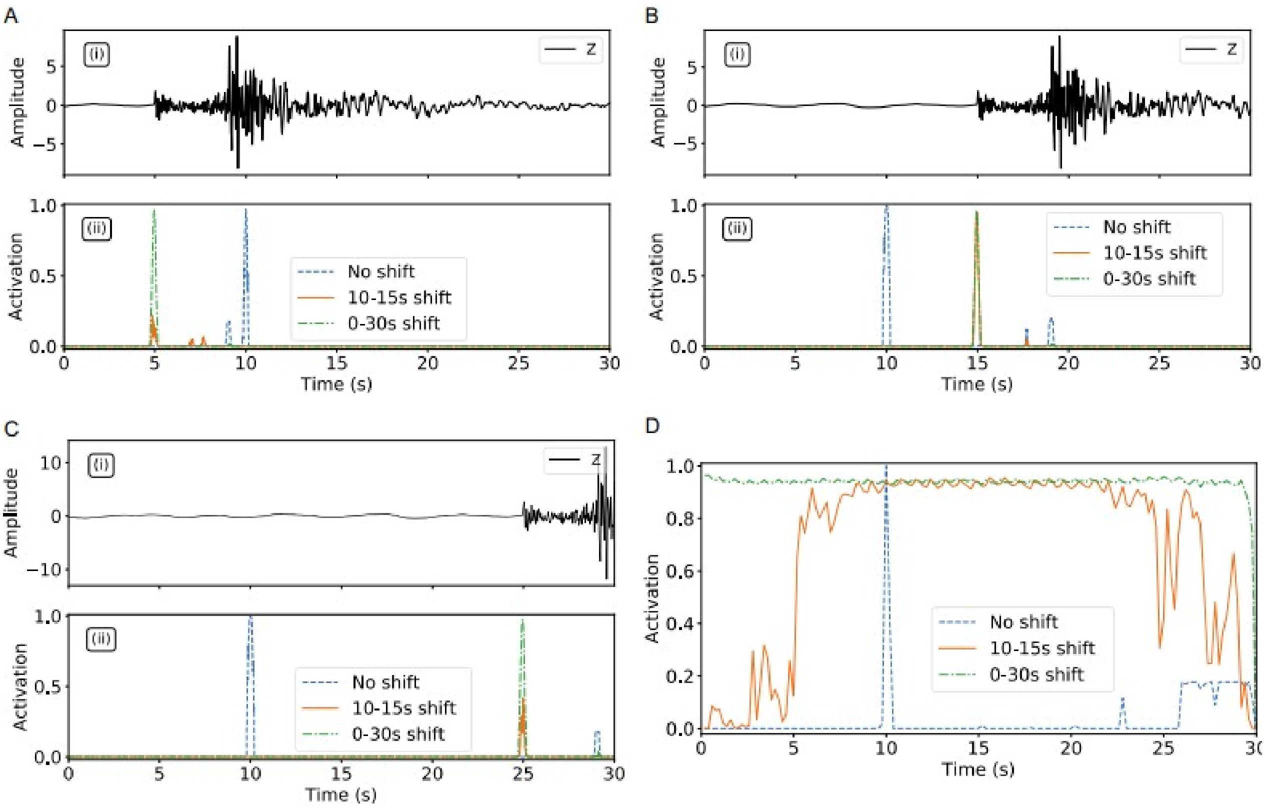
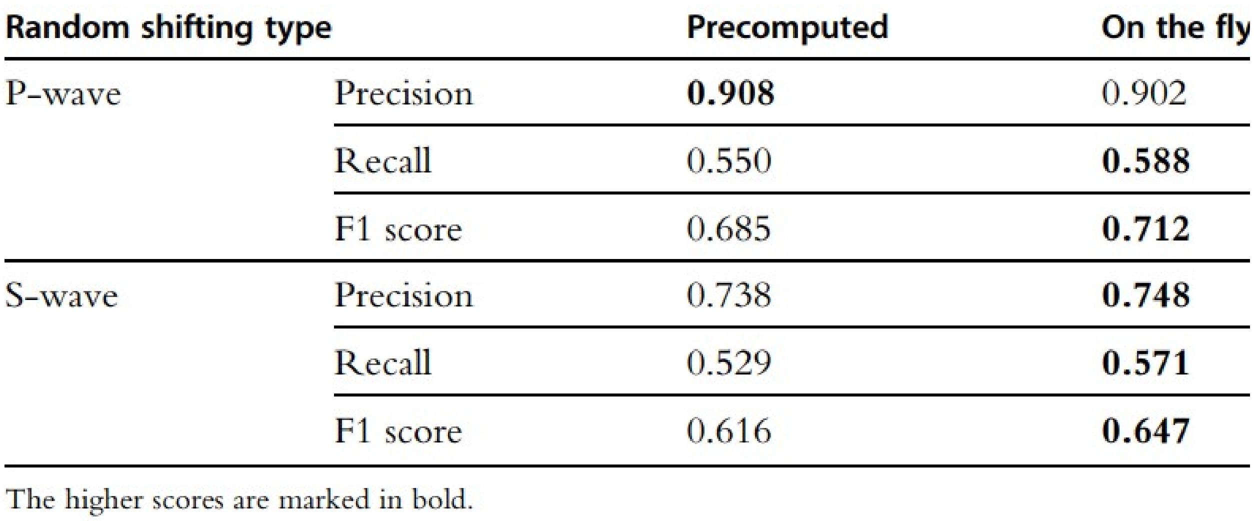


Figure 34 Three models' activation scores after being trained with various random shift augmentations. The estimated likelihood repercussions of neural networks trained with no random shift are shown in (A) through (C) with blue, random shift within 10-15s is illustrated with orange, and random shift within 0-30s is illustrated in green. The instances with P-wave arrivals at 5, 15, and 25 s are depicted in (A) and (C) as the test waveform glides from the left to the right border of the window. When moving the waveform across the window, (D) displays the forecasts for P-wave arrival at each time point. The neural network learns a constant time despite the waveform's content after training without random shift. When applied to continuous waveforms, training with an unfinished shift among 10 and 15s causes activations to decay near the window's margins, resulting in missed detections.[189]



Random shifting type		Precomputed	On the fly
P-wave	Precision	**0.908**	0.902
	Recall	0.550	**0.588**
	F1 score	0.685	**0.712**
S-wave	Precision	0.738	**0.748**
	Recall	0.529	**0.571**
	F1 score	0.616	**0.647**

The higher scores are marked in bold.

Table 3 Compare two realizations of randomization. (a) Pre-compute a fixed randomization for training sample, (b) Compute randomizations on the fly so that each training sample has a different alter in each epoch.[189]

5.2.2 Overlapping occurrences

Since cataloged earthquakes typically happen in isolation, with the exclusion of seismic swarms and aftershock sequences, training samples only contain one earthquake, also known as an "event" in the training window. However, neural networks may develop a minor bias of anticipating only one event to occur inside the time frame and repressing the detection of smaller occurrences that are simultaneously present in the time window if they are trained exclusively on single event data. For semantic-segmentation based approaches[171,174], which are intended to detect every event in a temporal window, this bias can lead to missing occurrences. When information is rich and earthquakes happen at much shorter intervals than is typical, we want an adequately-trained neural network to execute well on typical earthquakes while also generalizing to extreme scenarios like earthquake swarms and artificial earthquakes. Artificially superimposing events in a fashion that simulates such circumstances and gets rid of the bias of a single occurrence being in each window is an efficient augmentation to deal with the case of several events occurring inside a short window. Superimposition, often known as "stacking" in seismological jargon, is simply the addition of two or more time series. We also use a random proportion among event amplitudes during superimposition, which improves the neural network's capacity to identify smaller earthquakes that happen around larger ones. One instance of two earthquakes that happen near together in time is shown in figure 35. The neural network model that was trained with stacked occurrences predicts the second smaller occurrence with high likelihood scores for both P and S waves, in contrast to the model that was trained without superimposed occurrences, which only recognizes the first major event and ignores the second smaller one (figure 35(iv,v)).

The link where one station records two earthquakes at the same time is avoided, although the occurrence waveforms in the real data may overlap completely. These are the occurrence waveforms that are often discarded during process by hand. Due to the dependence of neural networks' effectiveness on data used for training, these situations may lead to an increase in false positives for typical seismic waveforms. As a result, we prevent augmentation from inadvertently adding entirely masked occurrences to the training. Our findings imply that a well-trained model ought to be able to generalize to the overlapping waveform events. Stacking significantly more overlapping occurrences may be helpful in specific applications to accurately recreate an earthquake swarm. We may approximately estimate the end of the earthquake utilizing the time spanned by P and S arrival times as the majority of datasets don't contain information regarding the end of the waveform (or end of the earthquake coda). The use of measurements based on envelope functions, similar to those employed in coda magnitude estimate, offers an additional choice.

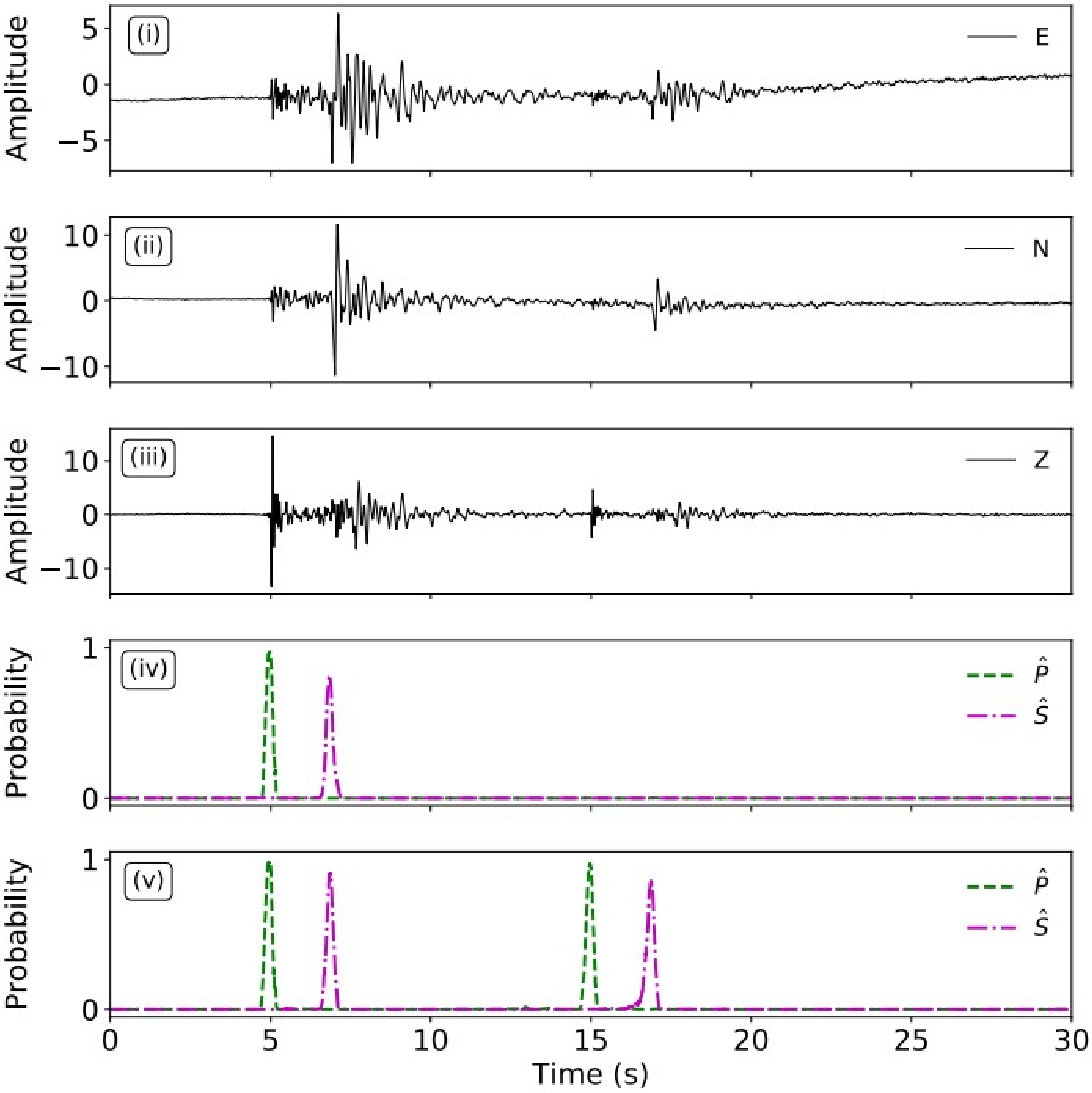


Figure 35 Waveforms of ENZ channels (i) through (iii), training lacking superimposed events (iv), training together with superimposed events (v), and anticipated activations when training without as well as with superimposed events are compared.[189]

5.2.3 Overlapping buzz

A strong neural network should be able to handle complex or low-quality data, even while the majority of manual labels are chosen from high-quality seismic data. The efficiency of neural networks deployed to low SNR data can be easily improved by superposing noise. The excellent credibility of labels from high SNR data may be maintained even when the waveforms are superimposed with heavy noise thanks to this enhancement, which is a clear benefit. The labels on the enhanced weak signals are more precise than those on the low SNR signals since they are de-amplified versions of well-known high SNR signals. We can modify the neural network's detection bounds by adjusting the ratio of the signal to the superposed noise. We can force the neural network to find weak signals buried in the background noise, in particular, by superimposing loud noise; yet, it should be emphasized that the likelihood of false positives may rise as well. However, as noise samples can be readily extracted from ongoing seismic recordings or from synthetically generated random noise, superposing noise is also a successful method of reducing overfitting on a small training dataset.

The neural network's effectiveness with and without superimposed noise is contrasted in figure 36. It is evident that even with a short training dataset, exceptional accuracy, recall, and F1 scores may be achieved on test samples with high SNR (>20 dB). When only excellent samples are utilized for training, recall is significantly lower for the low SNR samples tested (20dB). The recall and F1 score for the low SNR data are much higher after training with superimposed noise as an augmentation, while the efficiency for high SNR data is preserved.

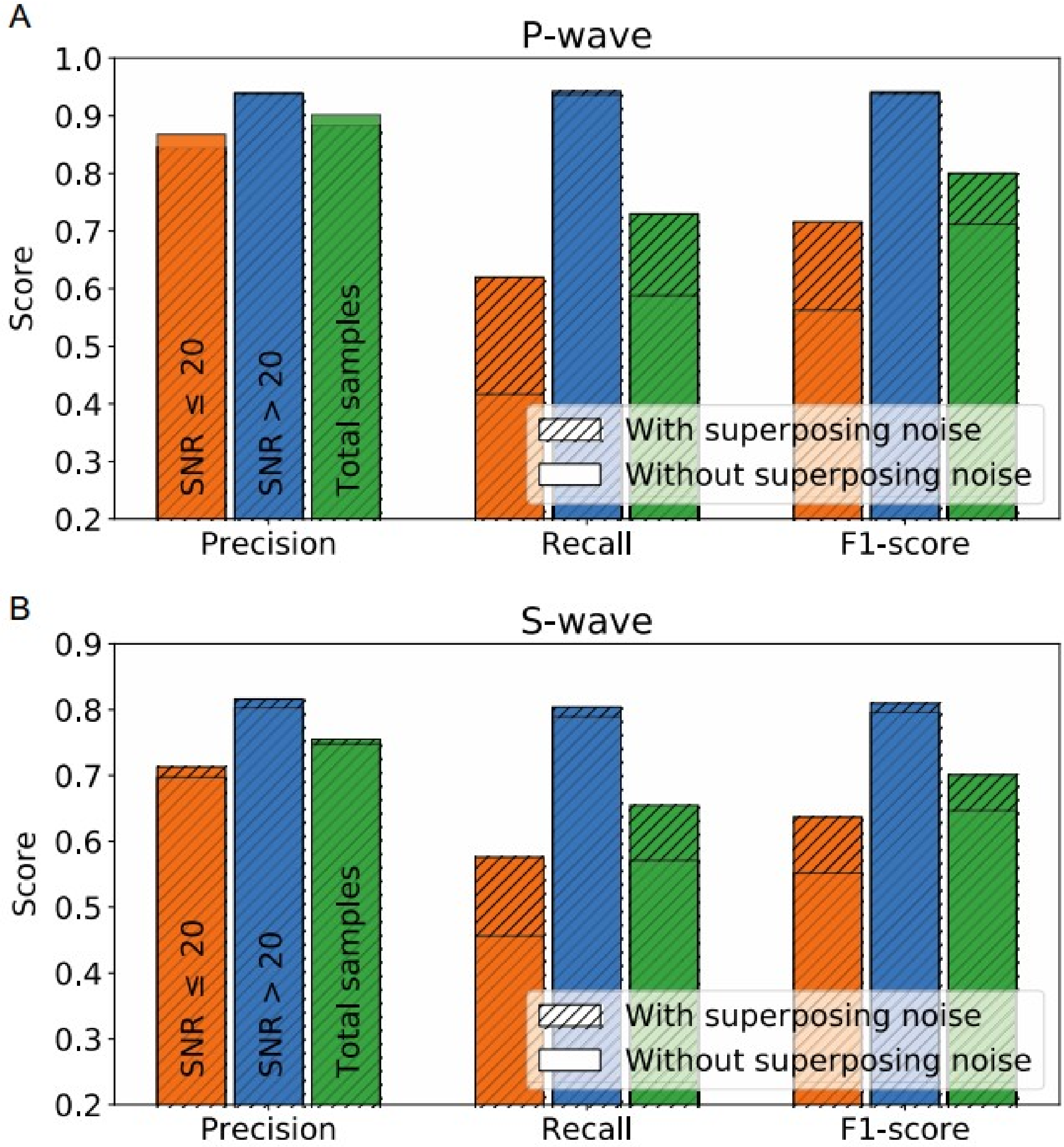


Figure 36 Performance comparison of detection for cases with and without noise superposition : Arrival of the P-wave (A) and the S-wave (B). Both models perform well for test data of high quality, such as SNR>20dB; yet, for test data of low quality, such as SNR 20dB, the model with enhancement greatly improves both the recall as well F1 score.[189]

It should be noted that the increase in recall is more important than the improvement in precision. This is due to the neural network trained with enhanced noisy data turns into more sensitive to weak signals submerged inside noise and recuperates a greater number of occurrences, but this could raise the possibility of false positives, limiting the improvement in precision. For purposes of comparison, it has been set the activation point threshold here. In real-world applications, we can adjust a series of activation thresholds, create a receiver's operating characteristic (ROC) curve, and identify an activation threshold that balances precision and recall. Depending on the setting, like deployments to borehole data, urban data, or ocean bottom seismometer (OBS) data, the choice of superposed noise will vary. Yet, care has to be taken to prevent including undetected occurrences within the noise windows, which at first would unintentionally boost the mis-labeling rate of the dataset and worsen performance. Employing real seismic noise collected by seismic instruments in situ ought to yield a more realistic enlargement and better performance. If the noise waveforms are not captured properly, there is also a chance of biasing the performance for a certain type of instrument. The outcome would be a less accurate depiction of real seismic noise, which is typically strongly coherent and non-stationary, even though the application of Gaussian noise ought to be safe in terms of preventing mislabeling. Actual seismic noise is therefore chosen for augmentation, provided that there is access to a trustworthy set of noise samples that have been examined for the presence of uncataloged earthquakes.

5.2.4 Noise that is falsely positive

As was previously demonstrated, superimposing noise on seismic signals enhances neural network performance for low SNR data. Another approach to dealing with complicated noise effects, like shaped pulses from urban vibration, is to provide falsely positive signals (non-earthquake signals). This lowers the rate of false positives and is especially useful for teaching neural networks to identify negative data. A small training dataset is capable of covering a certain range of noise since continuous seismic data contains complicated noise and nonstationary noise sources. As a outcome, the neural network may produce several false positives on undetected noise that has properties resembling seismic signals because it was trained on a small number of noise samples. To solve this issue, we can synthesize analogous non-earthquake signals or add these false positive noise samples to the training dataset for retraining or fine-tune the neural network and teach it features to identify these false positives to correct its predictions.

A typical scenario for the gathering of seismic data in which some data is missing because of instrument or telemetry failures is shown in figure 37. Due to the rapid modifications introduced, there may be false-positive results. If this form of noise is missing from the training dataset, deep learning algorithms may be confused by such kinds of false positives, which are frequent in classical approaches like STA/LTA. The model can make a mistaken prediction of P and S waves upon a sudden waveform adjustments, as illustrated in figure 37(iv), if it was trained without the proper augmentation. However, this kind of false-positive prediction can be efficiently suppressed by incorporating only a few of similar types of noise samples into the training data. Other typical forms of false positive noise, like impalsive signals from human activity, follow the same rationale. Due to the requirement for extensive both test and manual data examination, it might be difficult to distinguish between distinct kinds of typical false positives in practice. Active learning[192,193,194] may be a solution to increase the effectiveness of identifying false positive samples. Active learning seeks to build a system to rank these unlabeled data and annotate the most instructive samples first, like samples with the biggest uncertainty, because manually identifying false positives through a great amount of unlabeled samples during application is frequently challenging. By using these samples in training, learning effectiveness can be increased because they are more likely to be identified as false positives or false negatives.

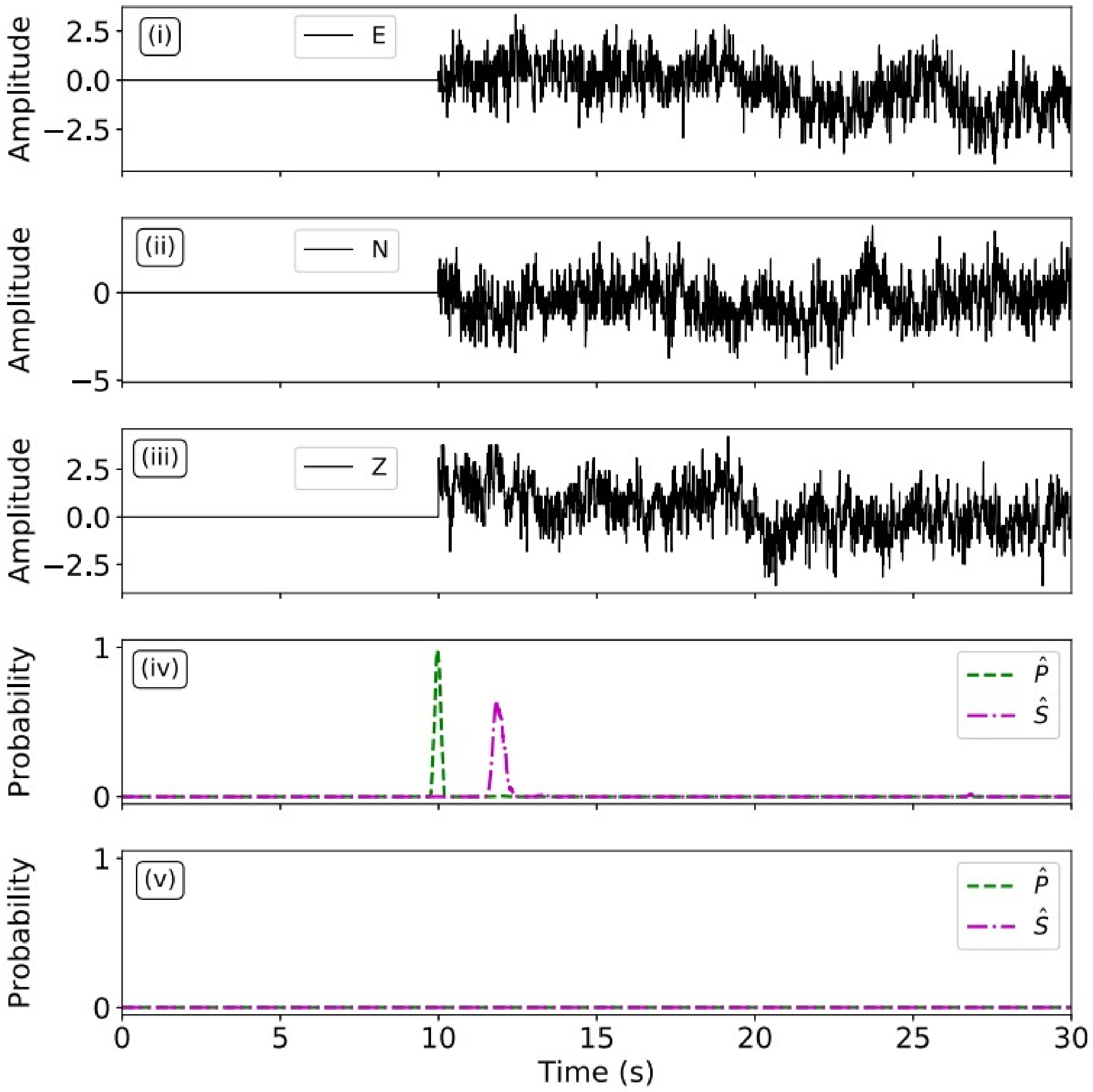
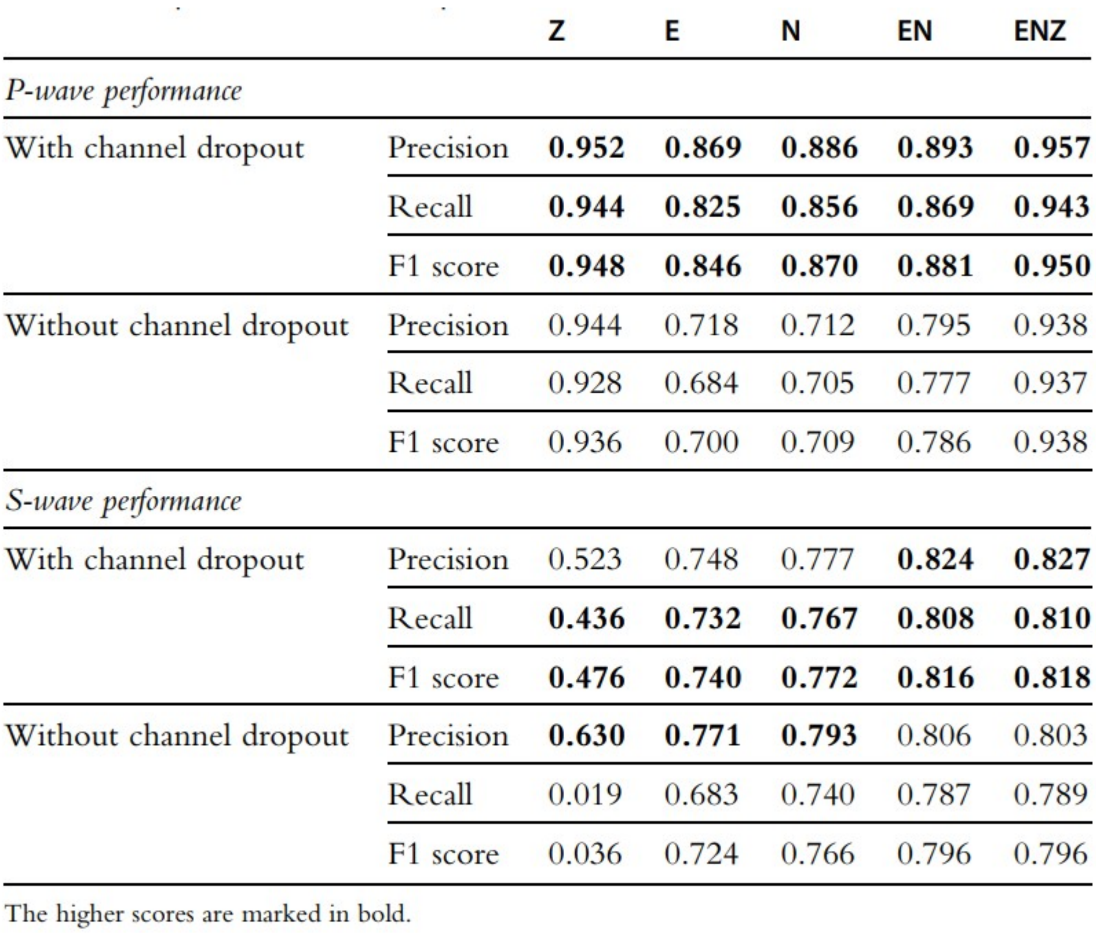


Figure 37 Comparison of predicted activations prior to and following the addition of false positive noise: (i-iii) ENZ channel waveforms; (iv) training lacking false positive noise; (v) training including false positive noise.[189]

5.2.5 Dropout channel

Modern earthquake seismology uses three-component seismic data the most frequently, yet single channel records predominate several historical documents and are still employed in some deployments. Additionally, it is not unusual for one of the channels in three-component recordings to fail as a result of telemetry or instrument faults. The efficacy of models trained on single-channel data employing three component data can be enhanced by augmentation. A strategy resembling dropout is an appropriate strategy[195,196]. It is dropped a couple of channels off the EZN input channels at random in the input layer. This channel discharge trains the neural network to predict data with lacking channels as well. It may be used a similar method to randomly dropping data from some of the stations during training for applications such as phase association, wherein the training is carried out using data from many stations[197]. By doing so, the neural network's robustness is increased in the unavoidable situations where data from certain stations is lost or damaged, preventing the network regarding overfitting on dominating stations.

The performance of training regardless of channel dropout is contrasted in Table 4. On test data of high quality (SNR>20dB), we employ the trained model and evaluate its performance across many components. Nevertheless, the model trained alongside channel dropout performs better with single-component data. Both models exhibit performance alike to that on three-component data.



		Z	E	N	EN	ENZ
P-wave performance						
With channel dropout	Precision	**0.952**	**0.869**	**0.886**	**0.893**	**0.957**
	Recall	**0.944**	**0.825**	**0.856**	**0.869**	**0.943**
	F1 score	**0.948**	**0.846**	**0.870**	**0.881**	**0.950**
Without channel dropout	Precision	0.944	0.718	0.712	0.795	0.938
	Recall	0.928	0.684	0.705	0.777	0.937
	F1 score	0.936	0.700	0.709	0.786	0.938
S-wave performance						
With channel dropout	Precision	0.523	0.748	0.777	**0.824**	**0.827**
	Recall	**0.436**	**0.732**	**0.767**	**0.808**	**0.810**
	F1 score	**0.476**	**0.740**	**0.772**	**0.816**	**0.818**
Without channel dropout	Precision	**0.630**	**0.771**	**0.793**	0.806	0.803
	Recall	0.019	0.683	0.740	0.787	0.789
	F1 score	0.036	0.724	0.766	0.796	0.796

The higher scores are marked in bold.

Table 4 Compare detection performance on different channels.[189]

The neural network's efficiency on single E-, N-, Z-, and EN-component permutations is informative and shows the knowledge it acquired to differentiate between P and S waves. Performance scores for selecting the P-wave arrival using only the Z component are comparable to those obtained utilizing all ENZ components, demonstrating that the Z component provides the majority of the data required for selecting P-wave arrivals. In contradiction, the crucial data for identifying S-wave arrivals is found in the horizontal EN components. This is consistent with how P and S waves are polarized, where P waves are more pronounced on the vertical component and S waves are more pronounced on the horizontal component.

5.2.6 Resampling Techniques

It can be difficult to effectively train DL models utilizing unbalanced datasets[198,199]. Due to the imbalance in earthquake magnitude distributions, this problem may be particularly important while training a neural network using seismic data. The number of major earthquakes for training is much more constrained than the number of tiny earthquakes because of a power law relationship among earthquake magnitude and the amount of earthquakes[200]. Applications like size estimation utilizing neural networks are directly impacted by this imbalance[51]. A specific training set may have problems with distance, depth, position, tectonic setting, source mechanism, magnitude type, equipment type, and SNR. Networks for seismic monitoring might also differ greatly in terms of station coverage and structure. These imbalances can limit a model's capacity to generalize to a wider range of earthquakes after being trained on a particular dataset. This makes it vital to look at data qualities as a training dataset is being built. Based on these exploratory studies, a suitable resampling strategy can be created to solve potential imbalance issues within a dataset.

By undersampling the bulk class or oversampling the minority class during training, random resampling is a technique to address the imbalance problem. This prevents the class distribution from becoming biased toward a few particular classes and allows for better generalization by training on an evenly balanced sample distribution. Nevertheless, resampling may result in unfavorable side effects. Losing some of the training data and lowering the training size are the costs of undersampling the majority of classes. Extreme oversampling can lead to the neural network just memorizing a small number of minority samples with comparable magnitudes or from the same region, which is obviously counterproductive to generalization. Furthermore, oversampling might have restricted employment for big earthquakes. Large events are not only more uncommon than small ones, but also more complex. Large earthquakes typically show intricate rupture patterns that span several faults over time and space. In order to fully represent the range of large magnitude earthquakes, oversampling may not be sufficient. A more efficient way to raise the ratio and diversity of the minority samples would be to combine oversampling with the methods of augmentation mentioned above. Using more sophisticated techniques like SMOTE [201], ADASYN[202], and GAN[97], training samples from existing instances could be synthesized.

5.2.7 Augmentation for the generation of synthetic data

In some instances, semi synthetic training data can be produced using augmentation techniques. Examples of problems where the established fact (the training objective) is unknown and impractical to obtain by human labeling include the seismic denoising problem[129] along with seismic detection upon scanned analog-seismograms[203]. We can solve this issue by synthesizing input alongside to target pairs sourced from plentiful seismic waves via augmentation. For instance, Zhu, Mousavi, and Beroza[129] used high SNR earthquake signals and a collection of noise waveforms to create an accurate denoising mask that served as the training goal for neural networks. As an outcome, this enlargement supplied an adequate number of training samples, by randomly blending signal and noise through a random proportion while training. So, in contrast from the forward synthesizing process, the neural network is trained to acquire a difficult inverse process to distinguish signal and noise.

Here, it's demonstrated an other instance of recovery of a clipped seismic waveform. For moderate till large earthquakes captured on nearby feeble motion equipment, clipped waveforms are frequently seen[204,205]. It is unable to directly obtain training data from prior waveforms due to the station's inability to view the genuine unclipped waveform. Yet, by manually cropping those waveforms, it is possible to synthesize the training data of unclipped waveforms. As a result, the neural network's input data comprises the synthetically clipped waveforms, and the training objective is the genuine unclipped waveforms, allowing to plain acquire a large quantity of training data through enlargement. This augmentation, like denoising, has the benefit of being produced from a signal with well-known -unclipped- established fact and providing a precise training label. In this scenario, it is employed the same network architecture as in the previous situations, but it is assessed the waveform difference among the recovered and genuine unclipped waveforms using a mean squared error (MSE). The waveforms that were recovered employing the model of neural networks trained upon the synthetic clipped waveforms are illustrated in the figure 38.

For several employments, the issue of unidentified established fact can be resolved by using enlargement to synthesize training data. The concept is similar to that of creating training data via applied calculations on a certain mathematical model representing the physical aspect of the object; nevertheless the augmentation approach generates training data based on genuine earthquake waveforms, which is cost-effective and results in realistic samples, due to the previously mentioned method. When applied to actual seismic recordings, the trained model is able to generalize semi-synthetic data more well. The neural network effectively apprentice an inverse modeling through the synthetic training data to reveal the real signal of interest that underlying the synthetic data if it is considered the data production process to be a forward function. On the flip side, numerical simulations, like finite failure modeling of big complex earthquakes, could offer an outlet for training data in situations wherein not just only the label is lacking as well as the real data is additionally rare. In this situation, it may be possible to blend synthetic earthquake waveforms with real-world noise to provide training data for better detection of major earthquakes. Nevertheless employed to real seismic data, the model trained using simulation data might encounter a generalization problem. To reduce the generalization gap, model fine-tuning either transfer learning upon certain real seismic waveforms is going to be required. Many other computer vision methods, like adversial discriminative adaption of domain[206], can be utilized to bridge the domain gap among simulation both real word. The significance of big earthquakes provides compelling impetus for further research in this area.

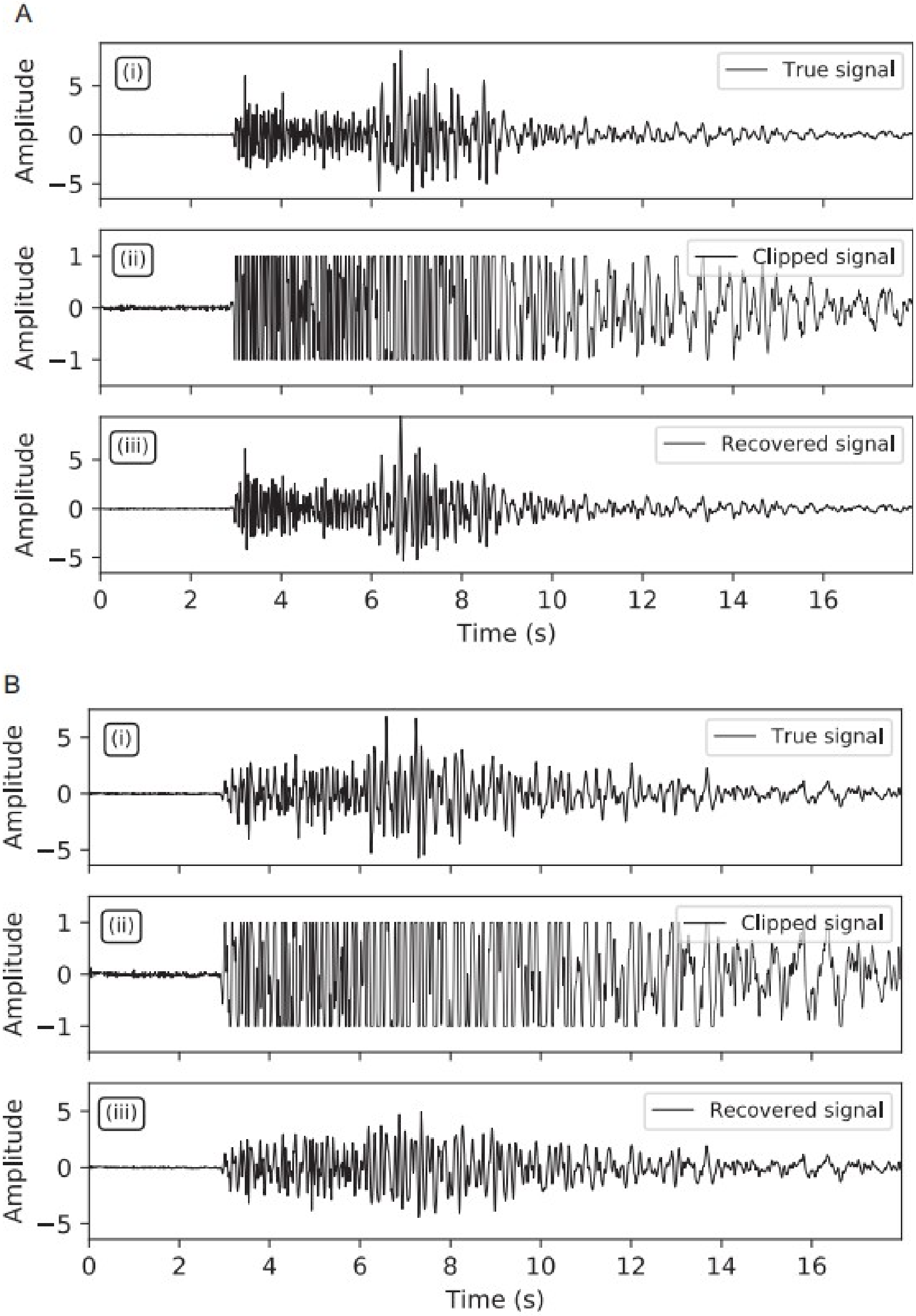


Figure 38 Two clipped waveform recovery instances (A, B): Real seismic signals (i), manually clipped seismic signals (ii), and retrieved seismic signals (iii) on the basis of neural networks trained on input-target pairings between (ii) and (i) [189]

6 Summary

From this review, is it clear that DL has a significant impact on traditional methods, bringing together as well as challenging both old and new ideas. Both DL and physical-based methods can offer great contributions to current and future studies[3]. It is shown how DL participates in the different geophysical domains, with a bigger interest to Seismology. What is its role in the science community. Additionally, are presented DL's tools for various applications on the filed. The main Libraries TensorFlow and Keras and how they affect the results, as well as the models based on these libraries and the widely used datasets. It's also presented when DL gives better results than experts and how it helps them to progress the methods and their efficiency. Furthermore, are discussed the possibilities that are turned into action in practical applications by DL and its tools. At the end, it is presented the data augmentation, that appears to improve the given outcomes and generalization of DNNs in different cases widely. And last but not least, it is shown how ML embraces the geoscience domain to bring optimized results as a helpful tool to scientists.

7 Bibliography-References-Web Sources

1. Some studies in machine learning using the game of checkers: Samuel, A.L., 1959
2. Machine learning: Mitchell, T.M.,1997
3. Deep learning for geophysics: Current and future trends: Yu S. & Ma J., 2021
4. Data-driven Geophysics: from Dictionary Learning to Deep Learning: Siwei Yu and Jianwei Ma
5. Deep learning for picking seismic arrival times: Wang, J., Xiao, Z., Liu, C., Zhao, D., Yao, Z., 2019
6. Earthquake phase arrival auto-picking based on u-shaped convolutional neural network. : Zhao, M., Chen, S., Fang, L., & David, A. Y., 2019
7. Hybrid event detection and phase-picking algorithm using convolutional and recurrentneural networks :Zhou, Y., Yue, H., Kong, Q., & Zhou, S.,2019
8. P wave arrival picking and first-motion polarity determination with deep learning:Ross, Z. E., Meier, M.-A., & Hauksson, E,2018
9. Earthquakegen: Earthquake generator using generative adversarial networks: Wang, T., Zhang, Z., & Li, Y. ,2019
10. Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking:Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C.,2020
11. Machine learning seismic wave discrimination: Application to earthquake early warning:Li, Z., Meier, M. A., Hauksson, E., Zhan, Z., & Andrews, J. ,2018
12. Reliable real-time seismic signal/noise discrimination with machine learning :Meier, M. A., Ross, Z. E., Ramachandran, A., Balakrishna, A., Nair, S., Kundzicz, P., et al. ,2019
13. CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection :Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C. ,2019
14. Deep learning models augment analyst decisions for event discrimination :Linville, L., Pankow, K., & Draelos, T. ,2019
15. Detection and classification of continuous volcano-seismic signals with recurrent neural networks :Titos, M., Bueno, A., García, L., Benítez, M. C., & Ibañez, J. ,2019
16. Volcano-seismic transfer learning and uncertainty quantification with Bayesian neural networks :Bueno, A., Benitez, C., De Angelis, S., Moreno, A. D., & Ibanez, J. M. ,2019
17. Automatic classification of volcano seismic signatures :Malfante, M., Dalla Mura, M., Mars, J. I., Metaxian, J. P., Macedo, O., & Inza, A. ,2018
18. Group invariant scattering. Communications on Pure and Applied Mathematics:Mallat, S.,2012

19. Locating induced earthquakes with a network of seismic stations in Oklahoma via a deep learning method :Zhang, X., Zhang, J., Yuan, C., Liu, S., Chen, Z., & Li, W.,2020

20. Phaselink: A deep learning approach to seismic phase association :Ross, Z. E., Yue, Y. S., Meier, M. A., Hauksson, E., & Heaton, T. H. ,2019

21. Seismic Tomography Using Variational Inference Methods :Zhang, X., & Curtis, A. ,2019

22. Machine learning approach to characterize the postseismic deformation of the 2011 Tohoku-Oki earthquake based on recurrent neural network :Yamaga, N., & Mitsui, Y. ,2019

23. Machine Learning Predicts Laboratory Earthquakes :Rouet-Leduc, B., Hulbert, C.,Lubbers, N., Barros, K.,Humphreys, C. J., & Johnson, P. A.,2017

24. Medium-term forecasting of loop current Eddy Cameron andEddy Darwin formation in the Gulf of Mexico with a divide-and-conquer machine learning approach.:Wang, J. L., Zhuang, H., Chérubin, L. M., Ibrahim, A. K., & Ali, A. M.,2019

25. Opportunities and obstacles for deep learning in biologyand medicine :Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P.,… Greene, C. S. ,2018

26. Quantum-chemical insights from deep tensor neural networks :Schutt, K. T., Arbabzadah, F., Chmiela, S., Muller, K. R., & Tkatchenko, A.,2017

27. Deep learning in medical image analysis :Shen, D., Wu, G., & Suk, H.-I. ,2017

28. druGAN : An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico :Kadurin, A., Nikolenko, S., Khrabrov, K., Aliper, A., & Zhavoronkov, A.,2017

29. Computing machinery and intelligence : Turing, A. M,1950

30. The hundred-page machine learning book: Burkov, Andriy,2019

31. Inversion of geophysical data by deep learning : Julio José Cárdenas Chapellín ,2020

32. Bayes, T. (1763). LII. An essay towards solving a problem in the doctrine of chances. By thelate Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton,AMFR S. Philosophical Transactions of the Royal Society of London: Bayes, T.,1763

33. Nouvelles methodes pour la determination des orbites des cometes:Legendre, A. M. , F. Didot.,1805

34. Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschiedrug ot druga. Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete:Markov, A. A. ,1906

35. Extension of the limit theorems of probability theory to a sum ofvariables connected in a Chain. Dynamic Probabilistic Systems:Markov, A. A.,1971

36. Υπολογιστική Νοημοσύνη: Α. Αλεξανδρίδης, Σημειώσεις 8ου εξαμήνου

37. Interpretable machine learning: definitions, methods, and applications :W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, Bin Yu,2019

38. Deep Learning with TensorFlow : Giancarlo Zaccone, Md. Rezaul Karim, Ahmed Menshawy

39. Seismic features and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression :Mousavi, S. M., Horton, S. P., Langston, C. A., & Samei, B.,2016

40. Deep learning of aftershock patterns following large earthquakes :DeVries, P. M. R., Viegas, F., Wattenberg, M., & Meade, B. J.,2018

41. Deep learning and process understanding for data-driven earth system science :Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat,2019

42. Machine learning for data-driven discovery in solid earth geoscience :Bergen, K. J., Johnson, P. A., de Hoop, M. V., & Beroza, G. C.,2019

43. Practical machine learning tools and techniques.Morgan Kaufmann.:Witten, I. H., Frank, E., & Hall, M. A.,2005

44. LIBSVM: A library for support vector machines. ACMTransactions on Intelligent Systems and Technology :Chang, C.-C., & Lin, C.-J. ,2011

45. Torch: A modular machine learning softwarelibrary.:Collobert, R., Bengio, S., & Mariethoz, J. ,2002

46. Automatic differentiation in PyTorch. :Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., …Lerer, A.,2017

47. Scikit-learn: Machine learning in Python. :Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O.,… Duchesnay, E. ,2011

48. Theano: A Python framework for fast computation of mathematical expressions :Theano Development Team.,2016

49. A Review on Conventional Machine Learning vs Deep Learning :Nitin Kumar Chauhan, Krishna Singh,2018

50. TensorFlow 2 Tutorial: Get Started in Deep Learning with tf.keras : Jason Brownlee

51. A machine-learning approach for earthquake magnitude estimation :Mousavi, S. M., & Beroza, G. C. ,2020

52. Bayesian-deep-learning estimation of earthquake location from single-station observations :Mousavi, S. M., & Beroza, G. C. ,2020

53. Very deep convolutional networks for large-scale image recognition:Simonyan, K., & Zisserman, A. ,2015

54. Deep residual learning for image recognition :He, K., Zhang, X., Ren, S., & Sun, J.,2016

55. Imagenet classification with deep convolutional neural networks :Krizhevsky, A., Sutskever, I., & Hinton, G. E. ,2017

56. Deep feature extraction and classification of hyperspectral images based onconvolutional neural networks :Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. ,2016

57. Target classification using the deep convolutional networks for SAR images :Chen, S., Wang, H., Xu, F., & Jin, Y. ,2016

58. Convolutional neural networks for large-scale remote-sensing image classification :Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. ,2017

59. Estimating ground-level PM2.5 by fusing satellite and station observations: Ageo-intelligent deep learning approach :Li, T., Shen, H., Yuan, Q., Zhang, X., & Zhang, L.,2017

60. Deep-learning tomography : Araya-Polo, M., Jennings, J., Adler, A., & Dahlke, T. ,2018

61. Unsupervised clustering of seismic signals using deep convolutional autoencoders :Mousavi, S. M., Zhu, W., Ellsworth, W., & Beroza, G. ,2019

62. Shale seismic facies recognition technology based on sparse autoencoder :He, Y., Cao, J., Lu, Y., Gan, Y., & Lv, S. ,2018

63. Data-driven super-parameterization using deep learning: Experimentation with multiscale Lorenz 96 systems and transfer learning :Chattopadhyay, A., Subel, A., & Hassanzadeh, P. ,2020

64. Parametric convolutional neural network-domain full-waveform inversion :Wu, Y., & McMechan, G. A.,2019

65. Near-real-time near-surface 3D seismic velocity and uncertaintymodels by wavefield gradiometry and neural network inversion of ambient seismic noise :Cao, R., Earp, S., de Ridder, S. A. L., Curtis, A., & Galetti, E. ,2020

66. A comparison of deep machine learning and Monte Carlo methods for facies classification fromseismic data :Grana, D., Azevedo, L., & Liu, M. ,2020

67. Improvement of a deep learning algorithm for totalelectron content maps: Image completion :Chen, Z., Jin, M., Deng, Y., Wang, J.-S., Huang, H., Deng, X., & Huang, C.-M. ,2019

68. Reconstruction of the basin-wide sea-level variability in the north sea using coastal data and generative adversarial networks :Zhang, Z., Stanev, E. V., & Grayek, S. ,2020

69. Discovering physical concepts with neural networks :Iten, R., Metger, T., Wilming, H., Del Rio, L., & Renner, R. ,2020

70. Deep learning :LeCun, Y., Bengio, Y., & Hinton, G. ,2015

71. The perceptron: A probabilistic model for information storage andorganization in the brain.:Rosenblatt, F.,1958

72. Learning while searching in constraint-satisfaction problems: Dechter, R.,1986

73. Neural networks and inversion of seismic data :Roth, G., & Tarantola, A.,1994

74. Location of subsurface targets in geophysical data using neural networks :Poulton, M., Sternberg, B., & Glass, C.,1992

75. Magnetotelluric inversion using regularized Hopfieldneural networks.:Zhang, Y., & Paulson, K. V.,1997

76. Neural network dynamic modelling of rock microfracturingsequences under triaxial compressive stress conditions :Feng, X.-T., & Seto, M. ,1998

77. Seismic attribute selection for machine-learning-based facies analysis :Qi, J., Zhang, B., Lyu, B., & Marfurt, K. ,2020

78. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising :Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. ,2017

79. Learning a deep convolutional network for image super-resolution :Dong, C., Loy, C. C., He, K., & Tang, X. ,2014

80. Dynamic routing between capsules. Advances in Neural Information Processing Systems :Sabour, S., Frosst, N., & Hinton, G. E. ,2017

81. Unsupervised representation learning with autoencoders: Makhzani, A.,2018

82. U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer assisted intervention : Ronneberger, O., Fischer, P., & Brox, T. ,2015

83. Unpaired image-to-image translation using cycle-consistent adversarial networks :Zhu, J., Park, T., Isola, P., & Efros, A. A.,2017

84. 70 years of machine learning in geoscience in review: Jesper Soren Dramsch

85. Long short-term memory :Hochreiter, S., & Schmidhuber, J. ,1997

86. Learning phrase representations using RNN encoder-decoder for statistical machine translation : Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. ,2014

87. Siamese neural networks for one-shotimage recognition :Koch, G., Zemel, R., & Salakhutdinov, R.,2015

88. Attention is all you need :Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., &Polosukhin, I.,2017

89. Learning multi-attention convolutionalneural network for fine-grained image recognition: Zheng, H., Fu, J., Mei, T., & Luo, J.,2017

90. P wave arrival picking and first-motion polarity determination withdeep learning:Ross, Z. E., Meier, M.-A., & Hauksson, E.,2018

91. Generalized seismic phase detection with deep learning:Ross, Z. E., Meier, M.-A., Hauksson, E., & Heaton, T. H., 2018,2018

92. Learning not to learn: Training deep neural networks with biased data:Kim, B., Kim, H., Kim, K., Kim, S., & Kim, J. ,2019

93. The inverse crime :Wirgin, A.,2004

94. Learning representations by back-propagating errors : Rumelhart, D. E., Hinton, G. E., & Williams, R. J,1986

95. Neural ordinary differential equations :Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. ,2018

96. Generative adversarial networks: An overview :Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. ,2018

97. Generative adversarial networks. Advances in Neural Information Processing Systems: Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.,2014, 2672–2680

98. A geometric understanding of deep learning :Lei, N., An, D., Guo, Y., Su, K., Liu, S., Luo, Z., et al.,2020

99. Joint inversion of seismic and electric data applied to 2D media :Garofalo, F., Sauvin, G., Socco, L. V., & Lecomte, I.,2015

100.Multimodal deep learning:Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y., 2011

101.Deep multimodal learning : A survey on recent advances and trends :Ramachandram, D., & Taylor, G. W.,2017

102. Decaf: A deep convolutional activation feature for generic visual recognition : Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., & Darrell, T. ,2014

103..............How transferable are features in deep neural networks. In Proceedings of the neural information processing systems : Yosinski, J., Clune, J., Bengio, Y., & Lipson, H.,2014

104............Learning and transferring mid-level image representations using convolutional neural networks : Oquab, M., Bottou, L., Laptev, I., & Sivic, J. ,2014

105........EVALUATION OF VARIOUS DEFINITIONS OF CHARACTERISTIC PERIOD OF EARTHQUAKE GROUND MOTIONS : Russell A Green, Jongwon Lee, Wanda Cameron, Alfredo Arenas,2011

106. Introduction to Seismology : Markus Båth,1979

107.. Non-real-time automated classification of seismic signals of slope failures with continuous random forests : Michaela Wenner, Clément Hibert, Alec van Herwijnen, Lorenz Meier, and Fabian Walter,2021

108. Seismic discrimination with artificial neural networks : Preliminary results with regional spectral data : Dowla, F. U., Taylor, S. R., & Anderson, R. W.,1990

109..Regional seismic event classification at the NORESS array: Seismological measurements and the use of trained neural networks: Dysart, P. S., & Pulli, J. J.,1991

110. Artificial neural network-based seismic detector :Wang, J., & Teng, T.-L.,1995

111. Detecting teleseismic events using artificial neural networks : Tiira, T.,1999

112. A neural network seismic detector : Madureira, G., & Ruano, A. E. ,2009

113...........................Seismic features and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression : Mousavi, S. M., Horton, S. P., Langston, C. A., & Samei, B.,2016

114..Supervised machine learning on a network scale: Applicationto seismic event classification and detection : Reynen, A., & Audet, P. ,2017

115.Seismic detection using support vector machines :Ruano, A. E., Madureira, G., Barros, O., Khosravani, H. R., Ruano, M. G., & Ferreira, P. M.,2014

116.Large margin classification using the perceptron algorithm : Freund, Y., & Schapire, R, 1999

117.Continuous earthquake detection and classification using discrete hidden Markov models :Beyreuther, M., & Wassermann, J.,2008

118.A machine learning approach for improving the detection capabilities at 3C seismic stations : Riggelsen, C., & Ohrnberger, M. ,2014

119. Page Rank for earthquakes : Aguiar, A. C., & Beroza, G. C.,2014

120...........Chances and limits of single-station seismic event clustering by unsupervised pattern recognition : Sick, B., Guggenmos, M., & Joswig, M.,2015

121.........Earthquake detection through computationally efficient similarity search : Yoon, C. E., O'Reilly, O., Bergen, K. J., Beroza, G. C., O'Reilly, O., Bergen, K. J., &Beroza, G. C. ,2015

122. Lomax, A., Michelini, A., & Jozinovi, 2019

123....CRED: A deep residualnetwork of convolutional and recurrent units for earthquake signal detection : Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C.,2019

124...Convolutional neural network for earthquake detection and location :Perol, T., Gharbi, M., & Denolle, M.,2018

125. Application of real time recurrent neural network for detection of small natural earthquakes in Poland :Wiszniowski, J., Plesiewicz, B. M., & Trojanowski, J. ,2014

126.Bayesian deep learning and uncertainty quantification applied to induced seismicity locations in the Groningen gas field in theNetherlands: What do we need for safe AI?:Gu, C., Marzouk, Y. M., & Toksoz, M. N.,2019

127. Machine learning and fault rupture: A review :Christopher X. Rena, Claudia Hulbertb, Paul A. Johnsonc,and Bertrand Rouet-Leduc,2020

128. Non-parametric seismic data recovery with curvelet frames :Herrmann, F. J., & Hennenfent, G.,2008

129. Seismic signal denoising and decomposition using deep neural networks :Zhu, W.,Mousavi, S. M., & Beroza, G. C. ,2019

130. Deep learning for denoising :Yu, S., Ma, J., & Wang, W. ,2019

131. White noise attenuation of seismic trace by integrating variational mode decomposition with convolutional neural network :Wu, H., Zhang, B., Lin, T., Li, F., & Liu, N. ,2019

132. Automatic velocity analysis using convolutional neural network and transfer learning :Park, M. J., & Sacchi, M. D. ,2019

133. Deep-learning inversion: A next-generation seismic velocity model building method :Yang, F., & Ma, J. ,2019

134.....Velocity model building in a crosswell acquisition geometry with image-trained artificial neural network :Wang, W., & Ma, J.,2020

135................Data-driven low-frequency signal recovery using deep-learning predictions in full-waveform inversion :Fang, J., Zhou, H., Elita Li, Y., Zhang, Q., Wang, L., Sun, P., & Zhang, J. ,2020

136. Deep learning for low-frequency extrapolation from multi offset seismic data : Ovcharenko, O., Kazei, V., Kalita, M., Peter, D., & Alkhalifah, T. ,2019

137.. Regularized elastic full-waveform inversion using deep learning :Zhang, Z., & Alkhalifah, T. ,2019

138..............Building long-wavelength velocity for salt structure using stochastic full waveform inversion with deep autoencoder based model reduction :Gao, Z., Pan, Z., Gao, J., & Xu, Z. ,2019

139.. The importance of transfer learning in seismic modeling and imaging :Siahkoohi, A., Louboutin, M., & Herrmann, F. J. ,2019

140.. Automatic velocity picking from semblances with a new deep-learning regression strategy: Comparison with a classification approach :Wang, W., McMechan, G. A., Ma, J., & Xie, F. ,2021

141. A theory-guided deep-learning formulation and optimization of seismic waveform inversion: Sun, J., Niu, Z., Innanen, K. A., Li, J., & Trad, D. O.,2020

142.. Multi-parameter full waveform inversions based on recurrent neural networks: Liu, S.,2020

143. Ml-descent: An optimization algorithm for full-waveform inversion using machine learning :Sun, B., & Alkhalifah, T. ,2020

144...........Building realistic structure models to train convolutional neural networks for seismic structural interpretation :Wu, X., Geng, Z., Shi, Y., Pham, N., Fomel, S., & Caumon, G., 2020

145...Fault seg 3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation :Wu, X., Liang, L., Shi, Y., & Fomel, S.,2019

146........Semiautomated seismic horizon interpretation using the encoder-decoder convolutional neural network :Wu, H., Zhang, B., Lin, T., Cao, D., & Lou, Y.,2019

147.Convolutional neural network for seismic impedance inversion :Das, V., Pollack, A., Wollner, U., & Mukerji, T.,2019

148.. An unsupervised deep-learning method for porosity estimation based on post stack seismic data :Feng, R., Mejer Hansen, T., Grana, D., & Balling, N.,2020

149.Shale anisotropy model building based on deep neural networks : You, N., Li, Y. E., &Cheng, A. ,2020

150.....Petrophysical properties prediction from prestack seismic data using convolutional neural networks :Das, V., & Mukerji, T.,2020

151. Seismic facies analysis based on deep convolutional embedded clustering : Duan, Y.,Zheng, X., Hu, L., & Sun, L. ,2019

152. Unsupervised seismic facies analysis via deep convolutional autoencoders:Qian, F., Yin,M., Liu, X., Wang, Y., Lu, C., & Hu, G. ,2018

153.Seismic impedance inversion based on cycle-consistent generative adversarial network :Wang, Y., Ge, Q., Lu, W., & Yan, X. ,2019

154......Convolutional recurrent neural networks based waveform classification in seismic facies analysis :Li, L., Lin, Y., Zhang, X., Liang, H., Xiong, W., & Zhan, S. ,2019

155.First-arrival picking with a U-Net convolutional network :Hu, L., Zheng, X., Duan, Y.,Yan, X., Hu, Y., & Zhang, X. ,2019

156.A robust first-arrival picking workflow using convolutional and recurrent neural networks :Yuan, P., Wang, S., Hu, W., Wu, X., Chen, J., & Van Nguyen, H.,2020

157. The SeisBench data format

158. Gradient flow in recurrentnets: The difficulty of learning long-term dependencies. A field guide to dynamical recurrentneural networks: Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J.,2001

159. Empirical evaluation of rectified activations inconvolutional network : Xu, B., Wang, N., Chen, T., & Li, M.,2015

160. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit: Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S.,2000

161.. Deep learning :Goodfellow, I., Bengio, Y., & Courville, A.,2016

162. Self-training with Noisy Studentimproves ImageNet classification:Xie, Q., Hovy, E.,Luong, M.-T., & Le, Q. V.,2019

163. Adam: A method for stochastic optimization:Kingma, D. P., & Ba, J. ,2014

164. On the importance of initialization and momentum in deep learning :Sutskever, I., Martens, J., Dahl, G., & Hinton, G.,2013

165. In 3D decision volume of SVM, random forest, and deep neural network : Dramsch, J. S., 2020

166. Python TensorFlow Big Data Analysis for the Security of Korean Nuclear Power Plants: Sangdo Lee , Jun-Ho Huh ,and Yonghoon Kim,2020

167. Keras – Installation

168. SeisBench - A Toolbox for Machine Learning in Seismology:Jack Woollam, Jannes Münchmeyer , Frederik Tilmann, Andreas Rietbrock, Dietrich Lange, ThomasBornstein, Tobias Diehl, Carlo Giunchi, Florian Haslinger, Dario Jozinovic, Alberto Michelini, Joachim Saul,and Hugo Soto,2021

169. Which picker fits my data? a quantitative evaluation of deep learningbased seismic pickers: Münchmeyer, J., Woollam, J., Rietbrock, A., Tilmann, F., Lange,D., Bornstein, T., Diehl, T., Giunchi, C., Haslinger, F., Jozinovic, D., Michelini, A., Saul, J., & Soto, H.,2021

170. Convolutional neural network for seismic phase classification,performance demonstration over a local seismic network:Woollam, J., Rietbrock, A., Bueno, A., & De Angelis, S.,2019

171. .Cred: A deep residual network of convolutional and recurrentunits for earthquake signal detection: Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C.,2019

172. .DeepPhasePick: a method fordetecting and picking seismic phases from local earthquakes based on highly optimized convolutional and recurrent deepneural networks:Soto, H. & Schurr, B.,2021

173. Generalized seismic phase detection with deep learning: Ross, Z. E., Meier, M.-A., Hauksson, E., & Heaton, T. H.,2018

174. Phasenet: a deep-neuralnetwork-based seismic arrival-time picking method: Zhu & Beroza, 2019

175. Instance–the italian seismic dataset for machine learning.:Michelini, A., Cianetti, S., Gaviano, S., Giunchi, C., Jozinovic,D., & Lauciani, V.,2021

176. Southern california earthquake data center.: SCEDC,2013

177. Seismic transfer program: SCEDC,2010

178. Leveraging deep learning in global 24/7 real-time earthquake monitoring at the national earthquake informationcenter: Yeck, W. L., Patton, J. M., Ross, Z. E., Hayes, G. P., Guy,M. R., Ambruz, N. B., Shelly, D. R., Benz, H. M., & Earle,P. S.,2021

179. Reliable real-time seismic signal/noise discrimination with machine learning: Meier, M.-A., Ross, Z. E., Ramachandran, A., Balakrishna,A., Nair, S., Kundzicz, P., Li, Z., Andrews, J., Hauksson,E., & Yue, Y.,2019

180.. .Reliable real-time seismic signal/noise discrimination with machine learningQ Meier, M.-A., Ross, Z. E., Ramachandran, A., Balakrishna, A., Nair, S., Kundzicz, P., Li, Z., Andrews, J., Hauksson, E., & Yue, Y.,2019

181. Data augmentation for improving deep learning in image classification problem: Mikołajczyk, A., & Grochowski, M.,2018 (pp. 117–122).

182. The effectiveness of data augmentation in image classification using deep learning: Perez, L., & Wang, J., 2017

183. Best practices for convolutional neural networks applied to visual document analysis: Simard, P. Y., Steinkraus, D., Platt, J. C., et al., 2003

184. Data augmentation for deep neural network acoustic modeling: Cui, X., Goel, V., & Kingsbury, B., 2015, p.p. 1469–1477.

185. Deep convolutional neural networks and data augmentation for environmental sound classification: Salamon, J., & Bello, J. P., 2017, p.p. 279–283.

186. Data augmentation for low-resource neural machine translation: Fadaee, M., Bisazza, A., & Monz, C., 2017

187. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification: Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H., 2018, p.p. 321, 321–331.

188. Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks: Um, T. T., Pfister, F. M., Pichler, D., Endo, S., Lang, M., Hirche, S., et al., 2017, pp. 216–220.

189. Seismic signal augmentation to improve generalization of deep neural networks: Weiqiang Zhu, S. Mostafa Mousavi, and Gregory C. Beroza, 2020, p.p. 151-174

190. Northern California earthquake data center: NCEDC. 2014, UC Berkeley Seismological Laboratory.

191. Bag of tricks for image classification with convolutional neural networks: He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M., 2019, pp. 558–567.

192. In Automatic earthquake detection by active learning: Bergen, K., & Beroza, G. C., 2017.

193. Improving generalization with active learning. Machine Learning: Cohn, D., Atlas, L., & Ladner, R., 1994, p.p. 201–221.

194. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning: Kirsch, A., van Amersfoort, J., & Gal, Y., 2019, pp. 7024–7035.

195. Dropout as a bayesian approximation: Representing model uncertainty in deep learning: Gal, Y., & Ghahramani, Z., 2016, pp. 1050–1059.

196. Dropout: A simple way to prevent neural networks from overfitting: Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R., 2014, p.p. 1929–1958.

197. An end-to-end earthquake monitoring method for joint earthquake detection and association using deep learning: Zhu, W., Tai, K. S., Mousavi, S. M., & Beroza, G. C., 2019.

198. Learning from imbalanced data: He, H., & Garcia, E. A., 2009, 1263–1284.

199. Handling imbalanced datasets: Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al., 2006, p.p. 25–36.

200. Frequency of earthquakes in California. Bulletin of the Seismological Society of America: Gutenberg, B., & Richter, C. F., 1944, p.p. 185–188.

201. SMOTE: Synthetic minority over-sampling technique: Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P., 2002, p.p. 321–357.

202. ADASYN: Adaptive synthetic sampling approach for imbalanced learning: He, H., Bai, Y., Garcia, E. A., & Li, S., 2008, pp. 1322–1328.

203. Earthquake detection in develocorder films: An image-based detection neural network for analog seismograms: Wang, K., Zhu, W., Ellsworth, W. L., & Beroza, G. C., 2019

204. An algorithm for detecting clipped waveforms and suggested correction procedures: Yang, W., & Ben-Zion, Y. 2010, p.p. 53–62

205. Restoration of clipped seismic waveforms using projection onto convex sets method: Zhang, J., Hao, J., Zhao, X., Wang, S., Zhao, L., Wang, W., et al., 2016, p.p. 39056.

206. Adversarial discriminative domain adaptation: Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T., 2017, pp. 7167–7176.